An Optically-Assisted 3-D Cellular Array Machine

Contract No. N00014-95-C-0094

Period of Performance: 1/31/95 to 4/30/97

Final Report

Presented to:

Office of Naval Research
Ballston Tower One
800 North Quincy Street
Arlington, VA 22217-5660

Technical Monitor:
Nick Bottka
(703) 696-4961

Presented by:

Physical Optics Corporation
Applied Technology Division
2545 W. 237th Street, Suite B
Torrance, California 90505
(310) 530-1416
CAGE NO. OAZ36

DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited

Principal Investigator:
Freddie Lin, Ph.D.

DTIC QUALITY INSPECTED 3

19970909 078

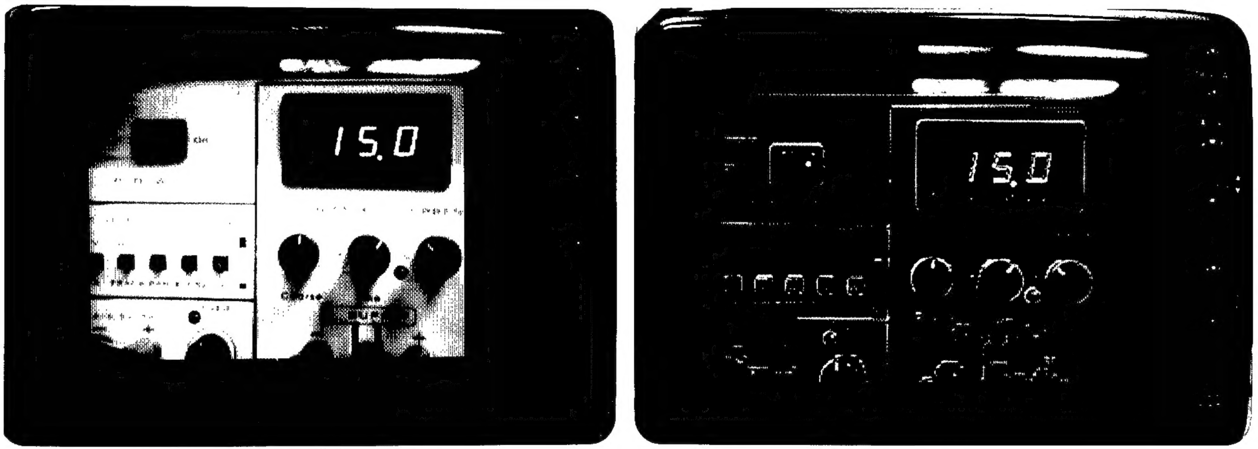
June 1997

TABLE OF CONTENTS

1.0 SUMMARY OF COMPLETED PROJECT

The goal of this SBIR Phase II project was to develop a cellular array machine for real-time image processing. In this Phase II project, we developed a *discrete-component* based Cellular Neural Network (CNN) circuitry, which can perform CNN based analog image processing, such as edge detection and image enhancement, in real time. This prototyping system performs 3 × 3 cellular cells, and is interfaced with a video camera input and a monitor output. Live video is captured with the video camera and input to this CNN prototyping system for data processing. The result is then shown in the monitor in real-time. Figure 1-1 shows the setup of our demonstration system, in which one CCD video camera, two discrete-component based CNN boards, and one video TV monitor were used. Figure 1-2 shows the experimental results for real-time image edge detection operation. These results show that our discrete-component CNN prototyping system is functioning as we expected and may have a potential to turn into a commercialized product.

Figure 1-1
Demonstration system setup.



(a) (b)

Figure 1-2
Experimental results: (a) original video image and (b) edge-detected video image.

2.0 INTRODUCTION

To increase the speed and image quality performance for next-generation image processing systems, we explored a new architecture -- a hybrid optically-assisted 3-D cellular array machine, as shown in Figure 2-1 in this project. The system consists of both analog and digital VLSI processing layers. These layers are stacked together in a 3-D configuration. Communication between layers is executed by optical interconnects that maintain high data communication throughput at the layer-to-layer level. The key function of the analog VLSI processing module is to perform initial coarse, high-speed image processing. The processed image data from the analog VLSI processing layer provides information to the digital VLSI processing layer for selective or prioritized fine image processing. In other words, the analog VLSI layer performs quick, coarse image processing operations on the received image, then directs the digital VLSI layer to the critical regions. These critical regions are low-contrast, high-clutter areas of the received image that may have been overlooked by the analog processing layer. In this way, the digital VLSI layer does not need to process every pixel in the received image, but can focus its processing power, flexible programming, and accurate calculations only on pixels located in the critical regions of the image.

The analog processing layer consists of an array of analog VLSI image processing nodes interconnected via a cellular array. Each node contains three parts: a photodetector array for object imaging; an analog VLSI processing circuit for high-speed, early image processing; and a photonic interface unit. The design of the photodetector array and analog VLSI processing circuit is similar to silicon retinas or early-vision neural chips. The function of the photonic interface unit is to convert electronic signals to optical signals for high-throughput, parallel, layer-to-layer communication. The analog VLSI layer transmits both processed image data (the control data for the digital VLSI) and unprocessed raw image data (image data for the digital VLSI module) to the digital VLSI layers.

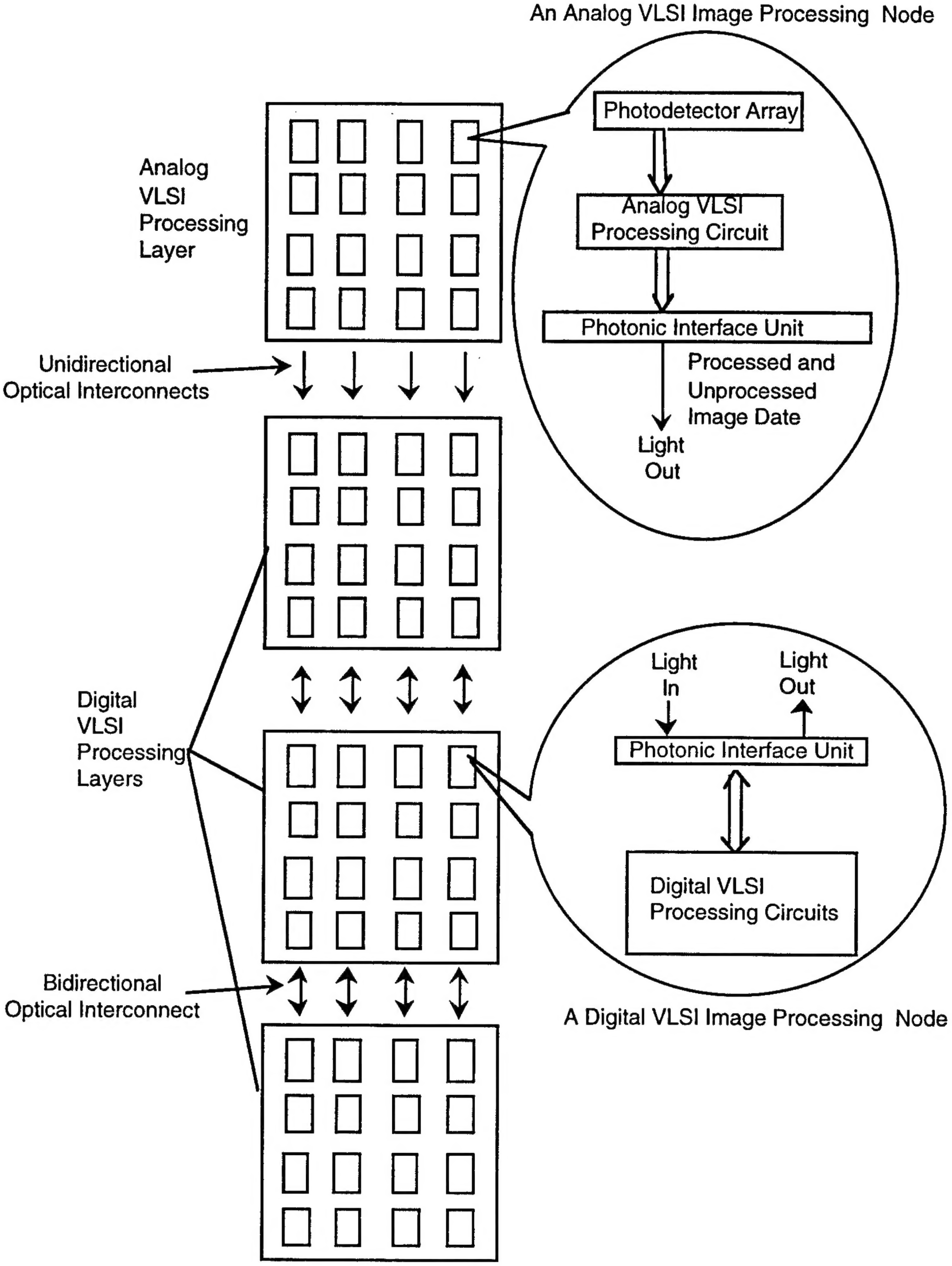

An Analog VLSI Image Processing Node
Photodetector Array
Analog VLSI Processing Circuit
Photonic Interface Unit
Processed and Unprocessed Image Date
Light Out
Analog VLSI Processing Layer
Unidirectional Optical Interconnects
Digital VLSI Processing Layers
Light In
Light Out
Photonic Interface Unit
Digital VLSI Processing Circuits
Bidirectional Optical Interconnect
A Digital VLSI Image Processing Node

Figure 2-1
Image processing architecture incorporating both analog and digital technologies. The analog layer is used for fast but coarse pre-processing, and the digital layer for high-precision, high-level processing.

The digital VLSI sections are arranged in a multi-layered cellular array configuration. The digital VLSI image processing nodes are interconnected by electronic wires within the layer and by optical interconnects between adjacent layers. Each digital VLSI image processing node consists of a photonic interface unit for bidirectional communication and a digital VLSI processing circuit for flexible, high-precision image processing.

Several layers of digital cellular arrays are required in the system for fine pre-processing and other high-level image processing. In other words, the analog cellular array layer performs quick, coarse image pre-processing of the received image and then identifies (for the digital cellular array layers) the coordinates of the critical regions that require further pre-processing. As a result, the digital cellular array layers can focus processing power, flexible programming, and high-precision computation on only those pixels in the critical regions of the image (see Figure 2-2). The digital processing technique can also be used to correct non-uniformity of the analog array unit and compensate for defective analog cells. This is of great importance for construction of large-scale analog cellular arrays, since problems of non-uniformity and defective cells become more severe as analog cellular array size increases. In addition to fine pre-processing operation, digital cellular array layers can also be used for high-level image processing operations, such as image segmentation, recognition, and analysis. Thus, the second layer of the system (the first layer of the digital cellular array) can be dedicated to fine pre-processing operations for critical regions, and the rest of the digital processing layers used for image segmentation, recognition, and analysis operations for the entire image.

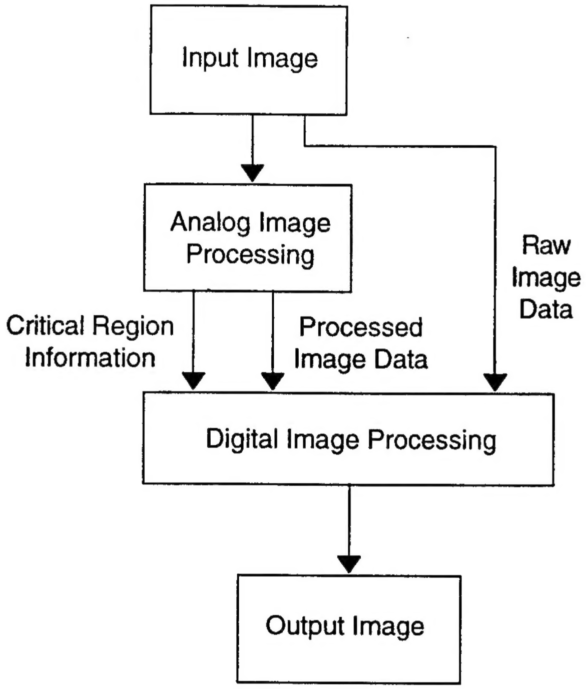


Figure 2-2
Flow chart of real-time image processing that uses both analog and digital processing techiques.

To achieve high-efficiency, high-throughput, and low-latency data communication between layers, as well as immunity to electromagnetic interference, an optical interconnect technique should be used. With an optical interconnect technique, electronic processing layers can be separated sufficiently for heat dissipation and crosstalk elimination, while still maintaining a high data communication bandwidth and parallel interconnections between layers. This is because the data bandwidth of the optical interconnect paths is independent of the separation between the layers.

In this project, POC focused on the development of an analog real-time image processing system, and implemented a cellular neural networking (CNN) system with discrete electronic components that performs only neighborhood connection.

To demonstrate the performance of this CNN prototyping system, POC connected this system to a camera input and a monitor output for real-time testing. In addition, several templates, such as edge enhancement and detection, were also implemented. Experimental results showed that edge detection and enhancement could be achieved in real-time. With this system, the executable time for computational intensive image processing can be saved significantly. The following sections describe the key components of the 3-D array machine, with an emphasis on CNN circuitry.

Sections 3.0 and 4.0 discuss the CNN machine. The VLSI CNN chip design (Section 3.0) and discrete-component CNN system development (Section 4.0) are both presented. In addition to this CNN system, we also explored the photonic interface chip development, which can reduce electromagnetic and radio frequency interference, and provide larger fanouts at higher bandwidths as well, when compared to conventional electrical interconnect techniques. Section 5.0 shows development of the photonic interconnect system and its interface with digital image processors. Conclusions are presented in Section 6.0.

3.0 CELLULAR NEURAL NETWORK (CNN) MACHINE

The cellular neural network (CNN), first proposed by Professor Chua, et al., from the University of California at Berkeley, is a network that consists of a two-dimensional array of locally interconnected analog signal processors. It has emerged as a powerful paradigm of multi-dimensional, locally connected, nonlinear processor arrays. The key idea used in the CNN universal machine is the "dual computing" paradigm, which combines analog array processing with logic operations by incorporating distributed analog memory and programmability, called analogic computing. Unlike hybrid computing in the analogic model, there are neither A/D and D/A converters nor digital representation of analog numbers; all signals and operators are either analog or logic. For image processing applications, each pixel of the image to be processed is usually associated with one cell; therefore, the processing is fully parallel. Furthermore, as opposed to other neural network topologies, interconnections between the processing elements are local only. This makes the CNN suitable for short-latency data processing. Another important property of the CNN is its programmability. The CNN can perform different types of convolutions and correlations using a programmable kernel function with a finite spatial window. CNN, CISC (Complex Instruction-Set Computing), and RISC (Reduced Instruction-Set Computing) are compared in Table 3-1. This table shows that CNN has a higher data throughput rate with less clocking frequency.

Table 3-1 Comparison of CNN, RISC, and CISC [9]

	CISC	RISC	CNN
Implementation	Pure digital	Pure digital	Hybrid (digital + analog)
Instruction-set size	12 - 350	30 - 85	16 (estimated)
Cycle per instruction (CPI)	2 - 15	<1.0	Varied for logic instruction and analog instruction
Addressing modes	12 - 24	3 - 5	2
Memory type	Digital memory	Digital memory	Digital memory + analog memory
I/O	Limited numbers	Limited numbers	Large I/O numbers required
General all-purpose registers number	8 - 24	32 - 192	Depends on local cell number
Clocking frequency	20 - 100 MHz (gate-level)	50 - 270 MHz (gate-level)	1 - 20 MHz (module level)
Data rate (connections/sec)	10^8	10^9 - 10^{10}	10^{14} - 10^{15}

Paralleled array processors based on cellular neural networks (CNNs) are very useful in high-speed, real-time applications. CNNs are continuous- or discrete-time artificial networks that consist of a multi-dimensional array of processing elements. The processing elements are locally interconnected with their neighboring elements. By using pre-defined templates, many complex problems in signal/image processing and optimization can be solved by CNNs. A CNN is also a special type of analog nonlinear processor array that is comprised of identical, equally spaced, processing elements of two or more dimensions, which are interconnected directly to their nearest neighbors. They are continuous- or discrete-time networks with local-interconnected processing elements that perform a very simple synaptic operation. After proposing the CNN concept, it rapidly evolved as a powerful paradigm of multi-dimensional, locally connected, nonlinear processor arrays such as N-dimensional, triangular-, rectangular-, or hexagonal- grid array. CNNs cover a wide range of applications that are typically characterized by their spatial dynamics. One particular area concerns filtering for image processing. The approach involves creating templates capable of several image processing applications and developing special purpose algorithms, such as halftoning and character recognition.

3.1 Basic CNN Principle

As shown in Figure 3-1(a), the CNN universal machine is based on a two dimensional $n \times m$ rectangular array network, where n and m are the number of rows and columns. Each element C (i, j), $1 \leq j \leq n$, $1 \leq j \leq m$, in the array corresponds to a cell in the CNN and is interconnected with its neighbor elements. Nr (i, j) is defined as elements C (k, l), $1 <= k <= n$, $1 <= l <= m$ for which $|k-I| <= r$, and $|l-j| <= r$, r is the distance between elements C(I, j) and its farthest neighbor element. Figure 3-1(b) shows the $n \times m$ array processing element and r = 1.

The mathematical description of CNN can be stated as:

$$\dot{v}_{xij}(t) = -v_{xij}(t) + \sum_{C(k,l) \in N_r(i,j)} A(i,j;k,l)v_{ykl}(t) + \sum_{C(k,l) \in N_r(i,j)} B(i,j;k,l)v_{ukl}(t) + I_b \quad (1)$$

In this equation, the B template is input from another cell and the A template is feedback from the cell itself. With this equation, we can draw an equivalent circuit, as shown in Figure 3-1(b), in which the input electrical current is from the adjacent cell and feedback from itself.

The internal state signal and the output signal of the processing element are denoted by $v_x(i, j)$ and $v_y(i, j)$. There are two kinds of weights for the processing element C(i, j) to communicate with its neighboring Nr(i, j) (i.e., the feedback weights A(i,j; j,i) and A(j,i; i,j), and the feedforward weight B(i,j; k,l) and B(k,l; i,j)). Although the processing element C(i , j) is connected only to its neighboring elements, it can communicate indirectly with all other elements in the array processor. Figure 3-1(b) shows that the input signal to the state node of the processing element consists of the weighted sum of feedback signals, and a constant bias term I_b. The bias term is used to adjust the threshold value of the neuron. An integration operation is performed, followed by the summation state node. The output signal of the circuit is obtained by $v_y(i , j) = f(v_x(i , j))$, where the nonlinear transfer function can be an appropriate non-decreasing function y = f(x), provided that f(0) = 0, $f(-\infty) \rightarrow -1$ and $f(+\infty) \rightarrow +1$. A widely used nonlinear transfer function is the sigmoid function as given by

$$y = f(x) = \frac{1 - e^{-\lambda x}}{1 + e^{-\lambda x}}$$

where λ is proportional to the gain of the sigmoid function.

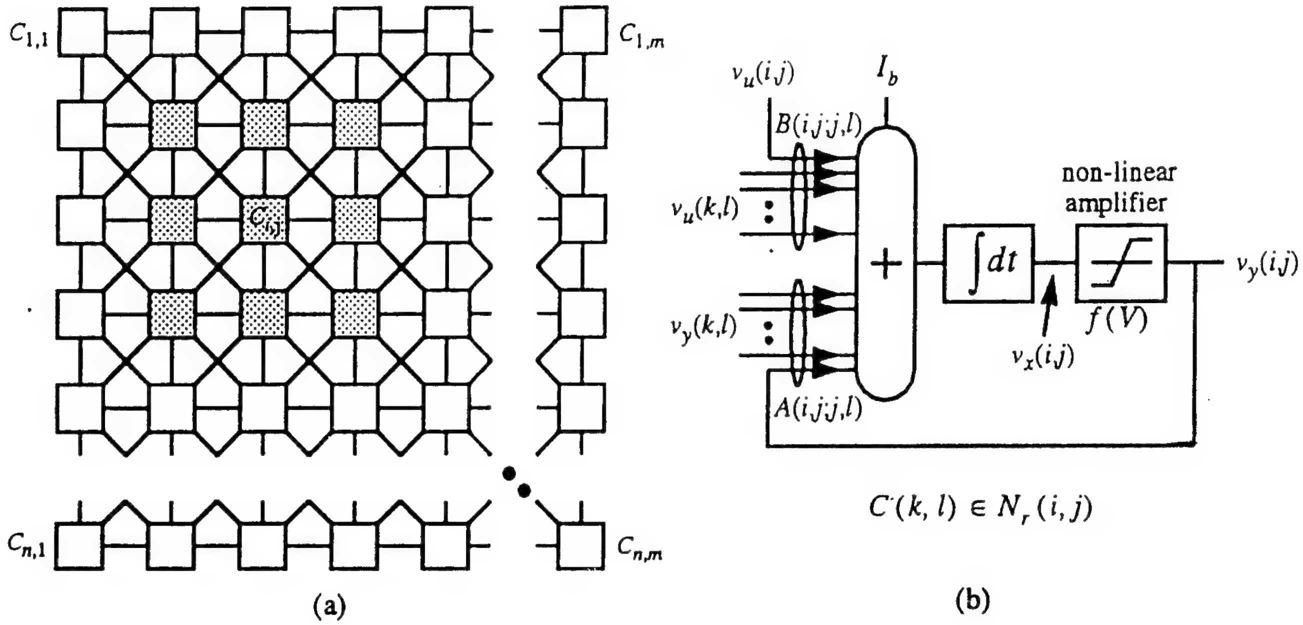


Figure 3-1
Cellular neural network (CNN): (a) An n-by-m cellular neural network on rectangular grid (shaded boxes are the neighborhood cells of C(i,j)); (b) Functional block diagram of neuron cell.

A sigmoid function with $\lambda = 2$ is shown in Figure 3-2. Even steady-state outputs must take binary values for cellular neural networks; the gain of the cell doesn't need to be large because the positive feedback factor in the network could be greater than one. Usually, a unity gain is used in the network. The sigmoid-like nonlinear function can be easily obtained by using integrated circuits, such as operational amplifiers in the voltage-mode operation. However, a piecewise-linear function is easier for mathematical analysis. Therefore, instead of a sigmoid function, we use a piecewise-linear transfer function as given by

$$y = f(x) = \frac{1}{2}\left(|x+1| - |x-1|\right) \tag{2}$$

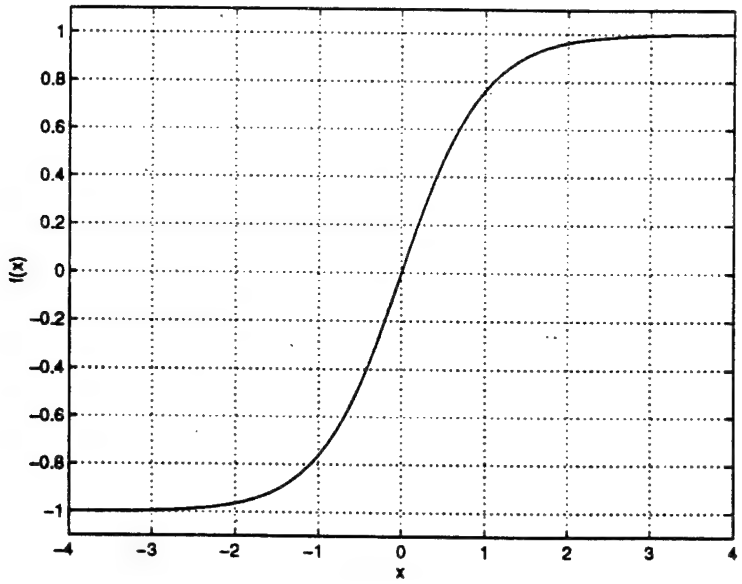


Figure 3-2
Sigmoid transfer function.

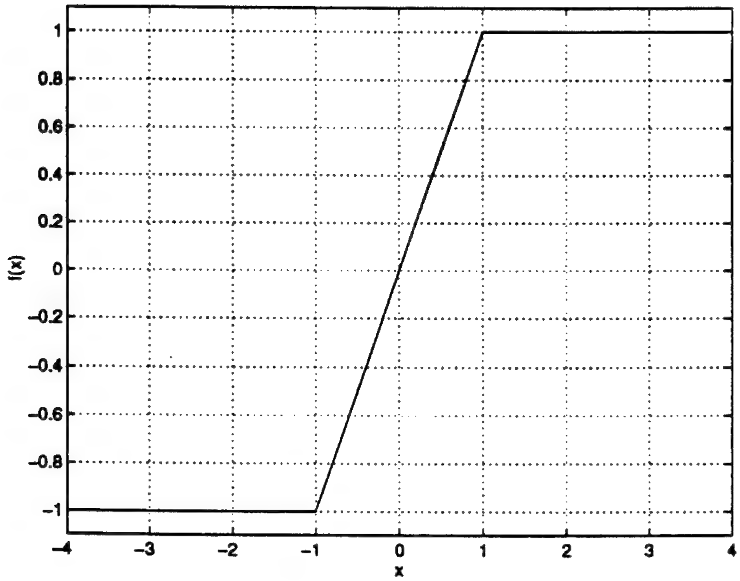


Figure 3-3
Characteristics of the piecewise-linear function.

Figure 3-3 shows the transfer characteristics of the piecewise-linear function. The feedback and feedforward weight of the network do not depend on the position of elements in the array processors, except at the edges, because the elements located on the edge have fewer neighbors than those inside the network. Special boundary elements with time-invariant state variables could be added on the parameters of the network. This property is a very attractive feature of the network for VLSI implementation of a large processor array. The interconnection weights of the network can be represented by (2r + 1) × (2r + 1) feedback and feedforward cloning templates.

$$\mathrm{T}_A = \begin{bmatrix} a_{-\mathrm{r},-\mathrm{r}} & a_{-\mathrm{r},-\mathrm{r}+1} & \cdots & a_{-\mathrm{r},\mathrm{r}} \\ a_{-\mathrm{r}+1,-\mathrm{r}} & a_{-\mathrm{r}+1,-\mathrm{r}+1} & \cdots & a_{-\mathrm{r}+1,\mathrm{r}} \\ \vdots & \vdots & \cdots & \vdots \\ a_{\mathrm{r},-\mathrm{r}} & a_{\mathrm{r},-\mathrm{r}+1} & \cdots & a_{\mathrm{r},\mathrm{r}} \end{bmatrix} \tag{3}$$

$$\mathrm{T}_B = \begin{bmatrix} b_{-\mathrm{r},-\mathrm{r}} & b_{-\mathrm{r},-\mathrm{r}+1} & \cdots & b_{-\mathrm{r},\mathrm{r}} \\ b_{-\mathrm{r}+1,-\mathrm{r}} & b_{-\mathrm{r}+1,-\mathrm{r}+1} & \cdots & b_{-\mathrm{r}+1,\mathrm{r}} \\ \vdots & \vdots & \cdots & \vdots \\ b_{\mathrm{r},-\mathrm{r}} & b_{\mathrm{r},-\mathrm{r}+1} & \cdots & b_{\mathrm{r},\mathrm{r}} \end{bmatrix} \tag{4}$$

By use of the vector and matric notations, Eq. (1) can be re-written as

$$C_x \frac{d\mathbf{x}}{dt} = -\frac{1}{R_x}\mathbf{x} + \mathbf{A}\mathbf{y} + \mathbf{B}\mathbf{u} + I_b\mathbf{w} \ , \tag{5}$$

where

$$\begin{aligned}
&N \quad = n \times m = \text{number of elements in the network,} \\
&\mathbf{u}_{N+1} = [u_1 \ \ u_2 \cdots u_N] = \left[\mathbf{v_{u1}}\middle|\mathbf{v_{u2}}\middle| \cdots \middle|\mathbf{v_{un}}\right]^{\mathbf{T}}, \\
&\mathbf{x}_{N+1} = [x_1 \ \ x_2 \cdots x_N] = \left[\mathbf{v_{x1}}\middle|\mathbf{v_{x2}}\middle| \cdots \middle|\mathbf{v_{xn}}\right]^{\mathbf{T}}, \\
&\mathbf{y}_{N+1} = [y_1 \ \ y_2 \cdots y_N] = \left[\mathbf{v_{y1}}\middle|\mathbf{v_{y2}}\middle| \cdots \middle|\mathbf{v_{yn}}\right]^{\mathbf{T}}, \\
&\mathbf{A}_{N\times N} = \mathit{toeplitz}\big((\mathbf{A_0}|\mathbf{A_1}|\cdots|\mathbf{A_r}|0|\cdots),(\mathbf{A_0}|\mathbf{A_{-1}}|\cdots|\mathbf{A_{-r}}|0|\cdots)\big) \ , \\
&\mathbf{B}_{N\times N} = \mathit{toeplitz}\big((\mathbf{B_0}|\mathbf{B_1}|\cdots|\mathbf{B_r}|0|\cdots),(\mathbf{B_0}|\mathbf{B_{-1}}|\cdots|\mathbf{B_{-r}}|0|\cdots)\big) \ , \\
&\mathbf{w}_{N\times 1} = [1\,1\cdots 1]^T.
\end{aligned} \tag{6}$$

Here,

$$\begin{aligned}
&\mathbf{v_{uk}}_{_{1\times m}} = [v_u(k,1) \ \ v_u(k,2)\cdots v_u(k,n)] \ , \\
&\mathbf{v_{xk}}_{_{1\times m}} = [v_x(k,1) \ \ v_x(k,2)\cdots v_x(k,n)] \ , \\
&\mathbf{v_{yk}}_{_{1\times m}} = [v_y(k,1) \ \ v_y(k,2)\cdots v_y(k,n)] \ , \\
&\mathbf{A_k}_{_{m\times m}} = \mathit{toeplitz}\big((a_{k,0} \ \ a_{k,1}\cdots a_{k,r}0\cdots),(a_{k,0} \ \ a_{k,\text{-}1}\cdots a_{k,\text{-}r}0\cdots)\big) \ ,
\end{aligned}$$

$$\mathbf{B}_{\mathbf{k}_{\mathrm{m\times m}}} = toeplitz\left(\left(b_{k,0}\ \ b_{k,1}\cdots b_{k,r}0\cdots\right),\left(b_{k,0}\ \ b_{k,\text{-}1}\cdots b_{k,\text{-}r}0\cdots\right)\right)\ . \tag{7}$$

The Toeplitz matrix, toeplitz(**a**, **b**), is defined as the matrix with **a** in the first row and **b** in the first column. The output vector **y** is confined within the N-dimensional hypercube because $-1 \le y_k \le +1,\ \forall k$. Thus, $\mathbf{y} \in \mathbf{D}^N = \left\{\mathbf{y} \in \mathbf{R}^N : -1 \le y_k \le 1;\ k = 1,2,\ldots,N\right\}$.

If A(i,j; k,l) = A(k,l; i,j) and B(i,j; k,l) - B(k,l; i,j), the cloning templates are called symmetric templates. In this case, A and B are symmetric matrices and the stability of the network is guaranteed. Actually, the symmetry of A is a sufficient condition for stability. The network always produces stable outputs in the steady state under the constraint conditions $\left|v_{\mathrm{x}}(i,j)(0)\right| \le 1$ and $\left|v_{\mathrm{u}}(i,j)\right| \le 1,\ \forall i,j$.

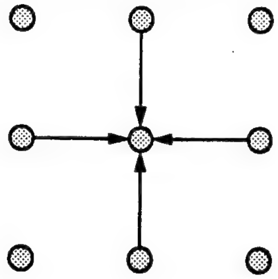
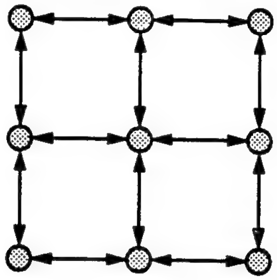
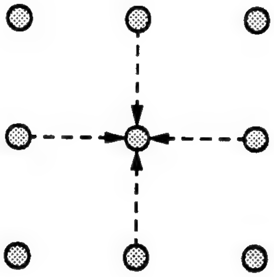
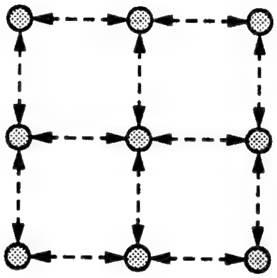
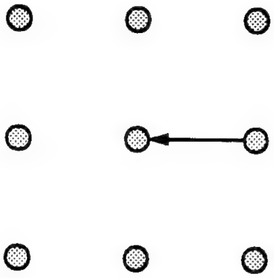
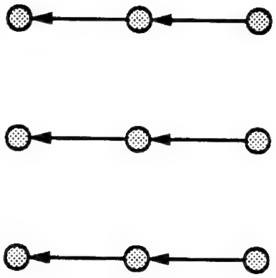
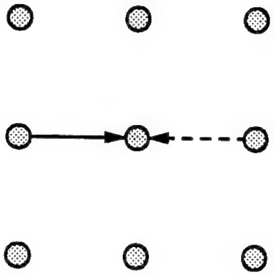
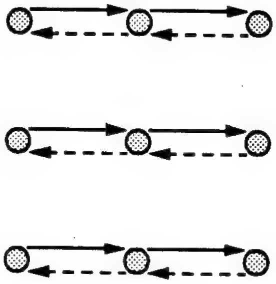
All the internal states Vx(i,j) Vt > = 0 in the cellular neural networks are bounded. The maximum state value v_{x},max can be determined by

$$\begin{aligned} v_{x,max} &= 1 + \mathrm{R_x}\left|\mathrm{I_b}\right| \\ &\quad + \mathrm{R_x} \max_{\substack{1\le \mathrm{i} \le \mathrm{n}, \\ 1 \le \mathrm{j} \le \mathrm{m}}} \left(\sum_{C(k,l)\in N_r(i,j)} \left(\left|A(i,j;k,l)\right| + \left|B(i,j;k,l)\right|\right)\right)\ . \end{aligned} \tag{8}$$

The terms in the right-hand side of Eq. (1) are contributed from initial value, bias signal, feedback, and feedforward integrations. In order to perform the summation and integration in the processing element, the operating range of the state node voltages must be at least $-v_{x,max} \le v_x(I,\, j) \le v_{x,max}$.

With this CNN circuit and the combination of A and B templates, various image processing operations can be achieved. Table 3-2 illustrates several possible A and B templates and corresponding image processing.

Table 3-2 Table of Templates, Template Graphs, and CNN Graphs

Name	A-template	Template graph	CNN graph
Hole filter	0.0 1.0 0.0 1.0 2.0 1.0 0.0 1.0 0.0		
Edge Detection	0.0 -0.5 0.0 -0.5 2.0 -0.5 0.0 -0.5 0.0		
Horizontal Shadow Detector	0.0 0.0 0.0 0.0 2.0 2.0 0.0 0.0 0.0		
Connectted Component Detector	0.0 0.0 0.0 1.0 2.0 -1.0 0.0 0.0 0.0		

3.2 CNN Chip Architecture Design

The global architecture of the CNN universal machine is shown in Figure 3-4. In this figure, each computing cell is controlled by a global analogic programming unit (GAPU). The GAPU receives analogic programs and executes them in a specified order, according to given template information. The block diagram of the GAPU is elaborated further in Figure 3-5, which shows where the analog program register (APR) stores several analog processing instructions. The logical program register (LPR) stores the local logic functions available for each cell. Different types of operations of programmable CNN universal cells are controlled by local switches. Various cell configurations are stored in the switch configuration register (SCR). Finally, the organization (i.e., the sequence of analogic instructions) is stored in the global analogic control unit (GACU). Figure 3-6 depicts the detailed information of an individual local cell. In this local cell, additional peripheral units are needed to achieve a universal cell. These peripheral components include local analog memory (LAM), local logic memory (LLM), the local communication and control unit (LCCU), the local logic unit (LLU), and the local analog output unit (LAOU).

In Figure 3-6, LAM1 is used for input, LAM2 for initial state and LAM3 for bias. LAMs(0~3) are general-purpose analog data registers for different applications. LAMs act as buffers which saves the driving loading of global signal output. By using local copies, LAM units and the LLM facilitate the implementation of several algorithm steps on the signal array. The SCR sends the switch configuration code to the LCCU; then the LCCU reconfigures the proper switches in the local cells. The LLU executes logic functions sent from the logical program register (LPR), and finally the LAOU converts analog value to logic value for further processing. Due to the huge amount of inputs and outputs in CNN architecture, the cells in the same rows can share bus line and I/O lines can be multiplexed. Therefore, the I/O bottleneck can be relieved.

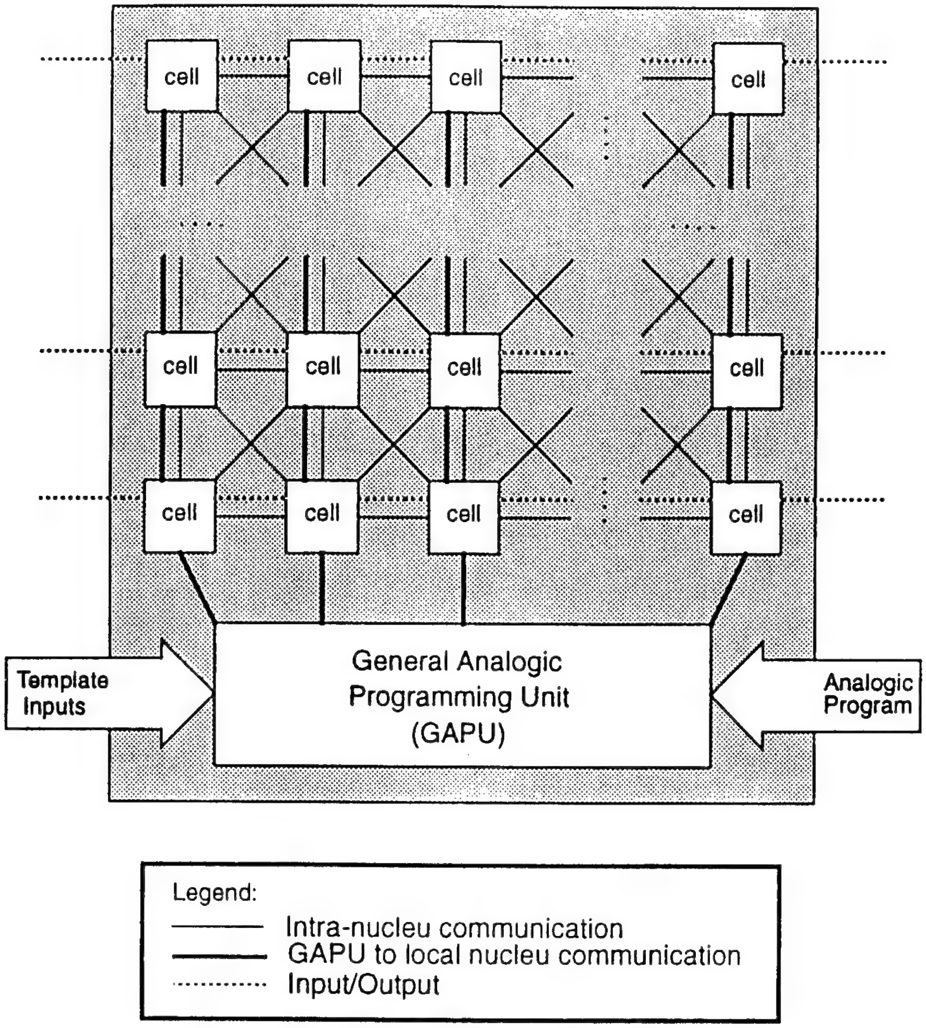

cell
cell
cell
cell
cell
cell
cell
cell
cell
cell
cell
cell
Template Inputs
General Analogic Programming Unit (GAPU)
Analogic Program
Legend:
Intra-nucleu communication
GAPU to local nucleu communication
Input/Output

Figure 3-4
Overall CNN universal machine architecture.

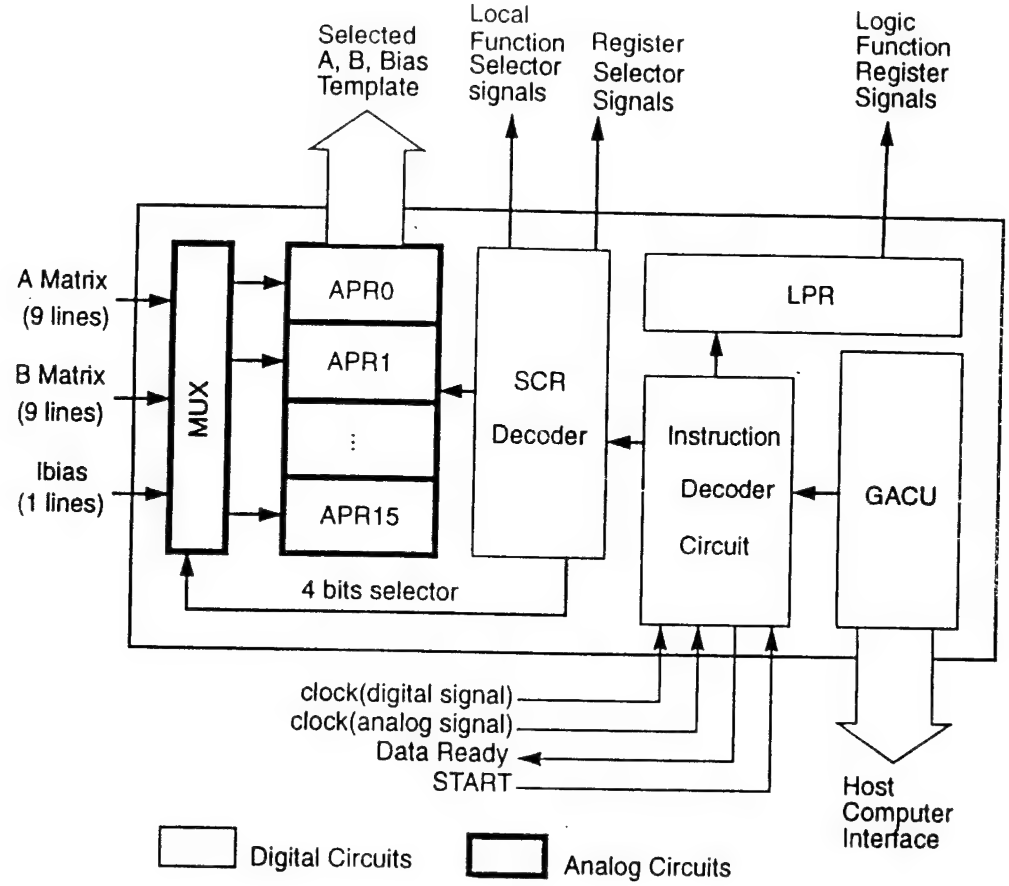


Figure 3-5
Block diagram of the GAPU.

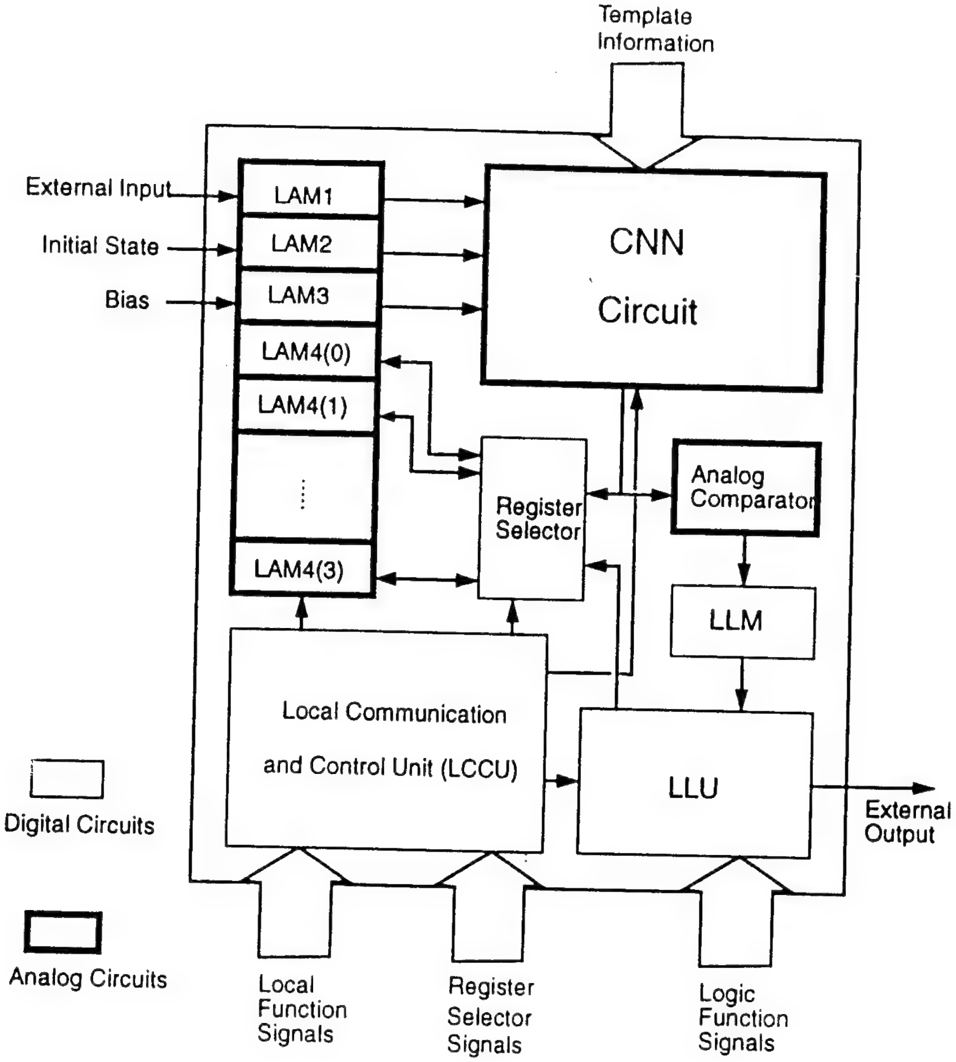


Figure 3-6
Block diagram of a local cell.

3.3 CNN Chip Schematic Design

Local interconnection and simple synaptic operators are desirable features of the CNN for VLSI implementation. However, a full programmable chip is still needed. To be fully programmable, 18 synapse weights described by feedback and feed-forward cloning templates, must be supported. Figure 3-7 shows a block diagram of one CNN processing element, consisting of a core cell, a synapse weight and input/output units. Four synapse circuits receive signals from the external input, self-feedback, and outputs of neighboring elements and multiply them with pre-stored template values. The resulting signals are sent to the core cell to perform summation, integration and nonlinear transformation. The output result is stored in a data latch and sent to the data bus by enabling the selection signal. In each operation, the initial state $V_x(0)$ must be initialized to a value between -1 and +1. By using control signals, an initial state $V_x(0)$ could share the same terminal with an external input signal. During the initialization operation, the control signal is low and the initial state can be placed on the capacitor C_x. At the same time, the outputs of synapse circuits are forced into a high-impedance state to avoid possible erroneous operation induced by the closed loop with parasitic capacitance at the state node.

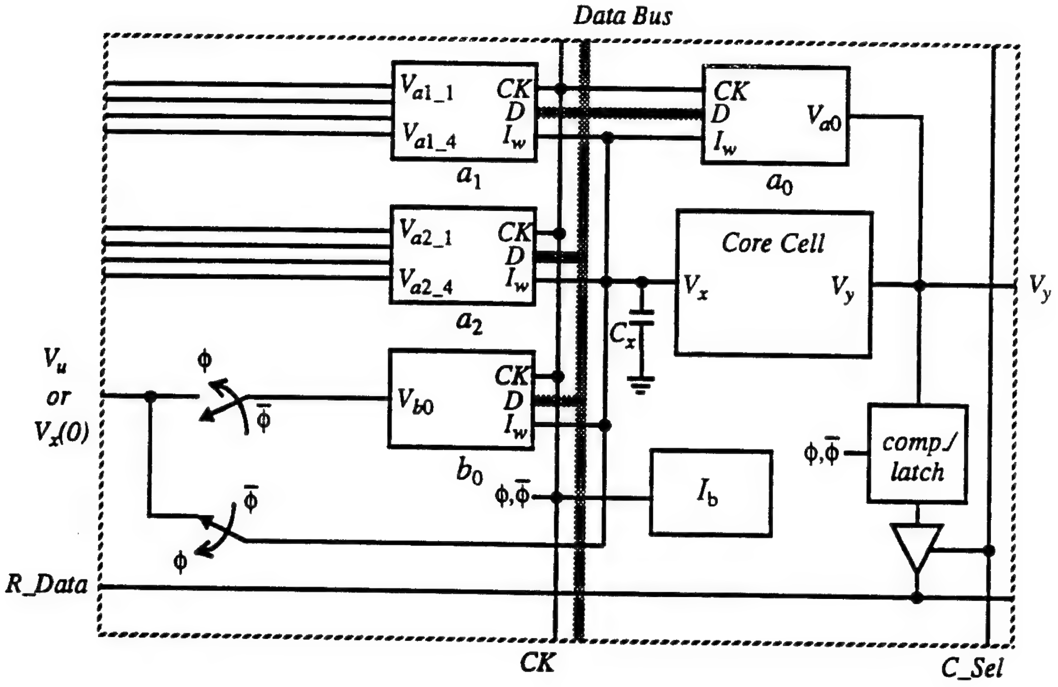


Figure 3-7
Block diagram of a single CNN processing element.

An n × m rectangular-grid array processor can be constructed by using the processing element shown in Figure 3-7 and appropriate interconnections with the neighboring element, as shown in Figure 3-8. Six 5-bit data registers are used to store the synapse weights a0, a1, a2, b0, b1, and b2. Those values can be transmitted to all processing elements through common control buses. To reduce the number of terminals for output signals, a multiplexing scheme can be used. Since a data bus is common to all elements in a row, the element outputs can be read out column by column through making the corresponding selection signal valid. The read operations can take place during the next network operation through the direct memory access (DMA), so network operation speed will not slow down in a moderate-size array.

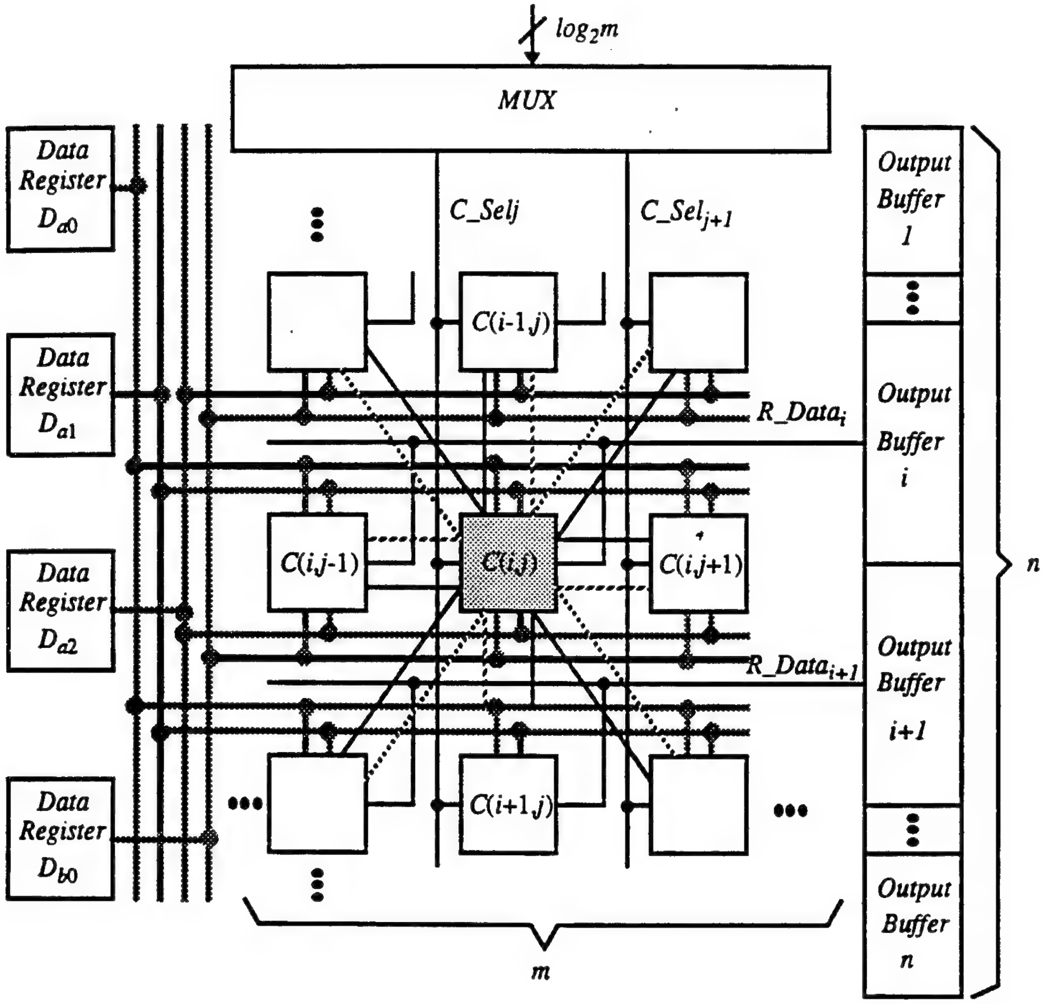


Figure 3-8
Architecture of an n × m CNN processor array.

3.4 Elemental Design in CNN Chips

The CNN cell consists of several basic functional blocks, including the current inverse circuit, voltage-to-current circuit, the piecewise-linear circuit, and the digitally-programmable synaptic weight circuit.

3.4.1 Voltage-to-Current Circuit

The voltage-to-current circuit converts external input voltage to current. It is a basic operational transconductance amplifier (OTA) circuit, as shown in Figure 3-9. Transistor sizes of the circuit have been carefully chosen according to the range of the input voltage and normalization current, to achieve the operating range (around 10 μA) of the circuit.

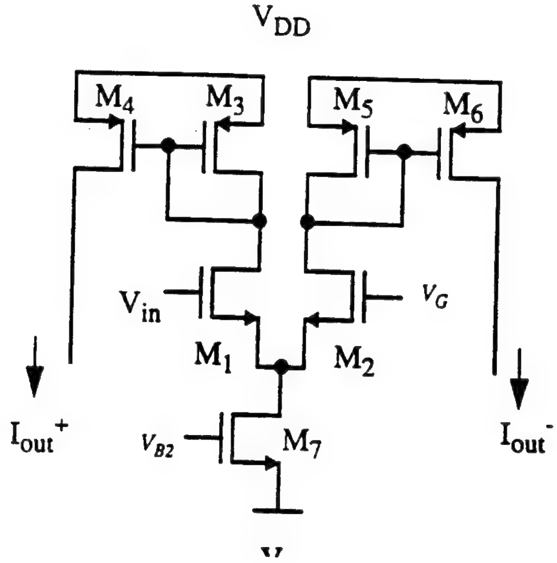


Figure 3-9
Schematic diagram of the voltage-to-current circuit.

3.4.2 Current-to-Voltage Circuit

A current-to-voltage circuit is shown in Figure 3-10. The output converts the summed current to voltage and performs several specific functions, such as thresholding, linear, amplification, and the sigmoid transfer function. Summation of currents, which are produced by weighted products, can be done by hardwiring, according to Kirchoff's current law. Current-to-voltage conversion is accomplished by comparing the input current with the reference current to produce output voltage V.

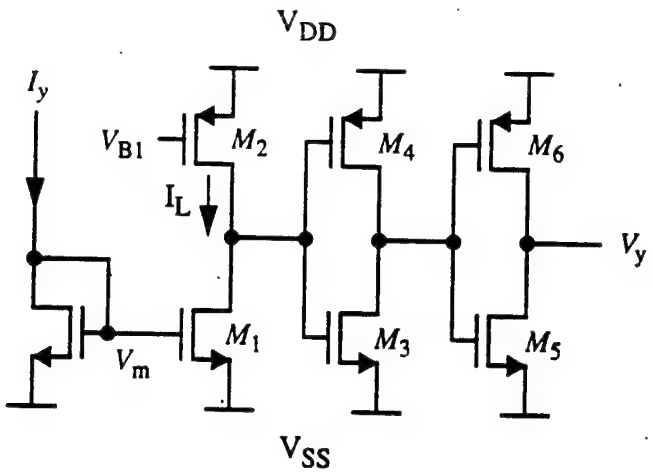


Figure 3-10
Schematic of a current-to-voltage circuit.

3.4.3 Bias Generation Circuit

Bias current is needed to adjust the threshold value of the neuron. One pMOS and one nMOS transistor can generate a bias circuit. A schematic diagram of the circuit is shown in Figure 3-11. The bias current is provided by controlling the gate voltage V_I of nMOS M1.

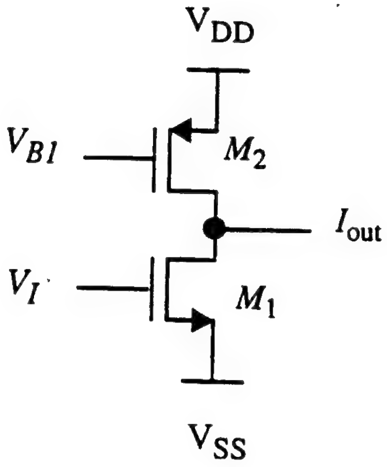


Figure 3-11
Schematic of bias generation circuit.

3.4.4 Current Inverse Circuit

Figure 3-12 shows a circuit schematic diagram of a current inverse circuit. It reverses the direction of the input current and provides constant input resistance with a value that is independent of the input current. Gate voltage V_B controls the equivalent resistance of the circuit.

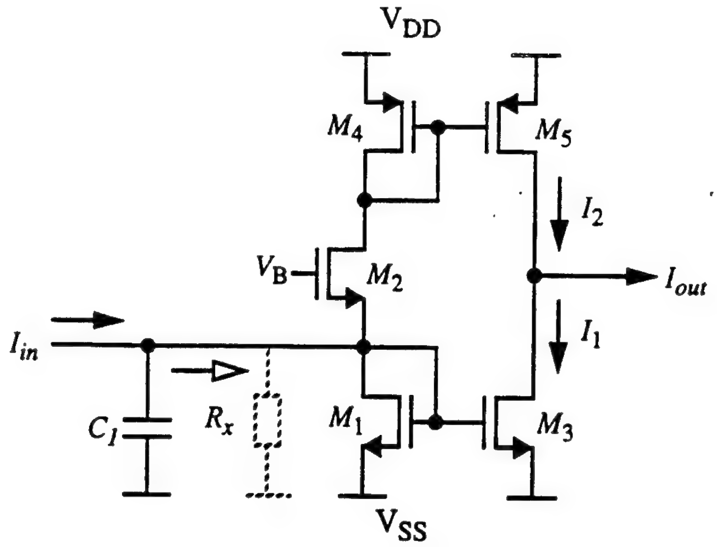


Figure 3-12
Schematic diagram of current inverse.

3.4.5 Piecewise-Linear Circuit

Figure 3-13 shows a schematic diagram of a piecewise-linear circuit, which is achieved by cascading two current limiters. If an input current is denoted by I_{in} and current flows through transistor M_5 is I_L, the limiting operation occurs at a positive value I_{in}=IL and negative I_{in} = -IL. When the input current I_{in} is less than -IL, the output current I_{out}+ - I_{out}- = -IL. When the input current I_{in} is greater than -IL but less than IL, the output current I_{out}+ - I_{out}- =I_{in}. When the input current is greater than IL, I_{out}+-I_{out}- = IL.

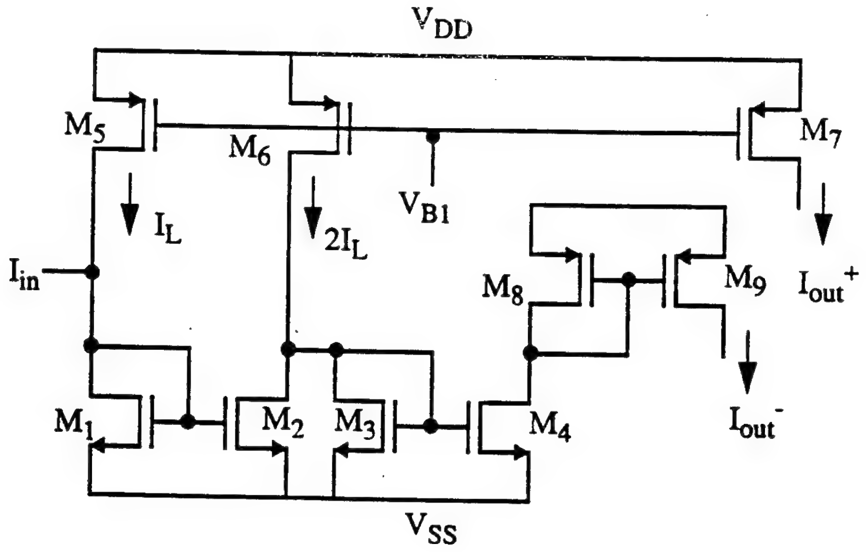


Figure 3-13
Schematic diagram of piecewise-linear circuit [16,35].

3.4.6 Digitally-Programmable Synaptic Weight Circuit

A digitally-programmable synaptic weight circuit is used to implement the cloning templates of the network. This circuit can perform four-quadrant multiplication.

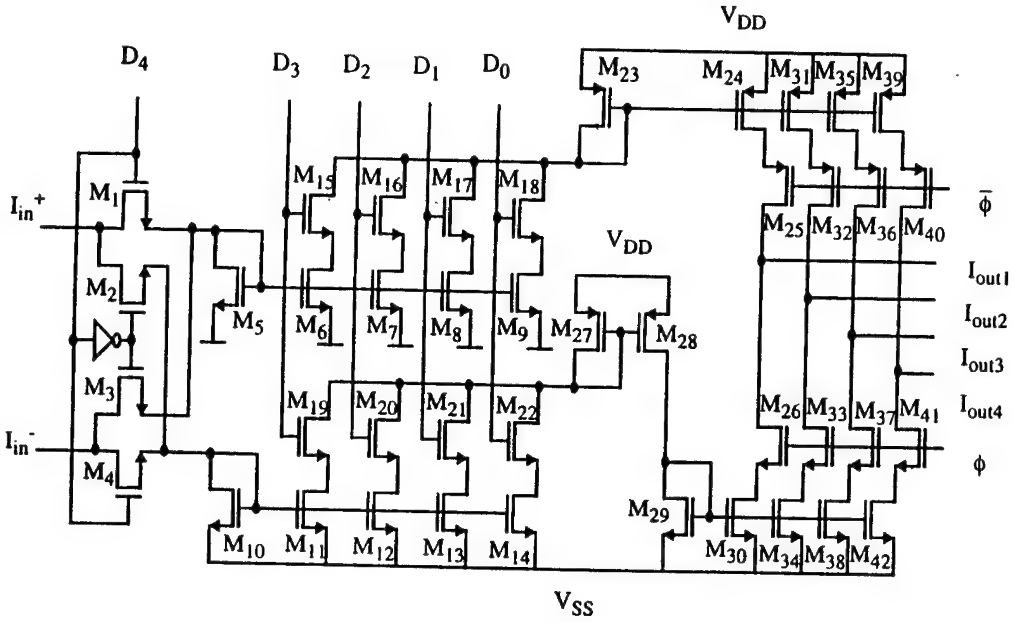


Figure 3-14
Digitally-programmable synaptic weight circuit [16,35].

3.5 Image Processing by CNNs Chips

The CNN paradigm showed great potential for image processing, such as noise removal, smoothing, hole filling, edge, corner, and shadow and motion detection. The processing speed is very high and the typical time required to become stable can be as small as 100 ns. With gray-scale image input, CNN can be applied to:

- motion estimation and detection
- collision avoidance
- halftoning, including motion compensation
- object counting, size estimation and path tracking.

To see how cellular neural networks implement image processing, we must first revise the differential equation (1) by approximating the derivation of x_{ij} through its corresponding differential equation:

$$C \cdot \frac{dx_{ij}(t)}{dt} = \frac{-1}{R} x_{ij}(t) + \sum A(i,j;k,l) y_{kl}(t) + \sum B(i,j;k,l) u_{kl} + I_b \tag{9}$$

where $y_{ij}(t) = \frac{1}{2}\left(\left|x_{ij}(t)+1\right| - \left|x_{ij}(t)-1\right|\right)$.

The above equation can be interpreted as a two-dimensional filter for transforming an image's pixel represented by $x_{ij}(t)$ into x(t+1) at next time step. In other words, this equation represents an image at time t which depends on the initial image $x_{ij}(0)$ and the dynamic rules of the cellular neural network. The filter is nonlinear because of the nonlinear output function of CNN. For the one-step filter in equation, the pixel value $x_{ij}(t+1)$ of an image are determined directly from the pixel values, $x_{ij}(t)$, in the corresponding neighborhood N(I, j). Therefore, a one-step filter can only make use of the local property of images. When global properties of a image are important, the above one-step filter can be iterated n times to extract additional information from the image. As mentioned before, each pixel is mapped onto a CNN cell, an image processing function in the spatial domain that can be expressed as g (i, j) = T(f(i, j)), where f(i, j) is the input image, g(i, j) is the processed image and T is the function on f(i, j) defined over the neighborhood of (i, j). For a CNN, this means that an output image cell is only influenced by input image pixels within some extended area in the neighborhood of the corresponding output image pixel. In a common image processing application, T(•) is usually carried out as a convolution process between a response function array and the input image. Which CNNs will serve the applications mentioned above depends on the values of the templates.

As mentioned before, CNN chips are very suitable for front-end, high-speed early vision processing; the weak point is precision. To compensate, we used a digital DSP chip to further process both the output from the CNN and the raw image data directly. As shown in Figure 3-15, a highly noisy, low contrast input is presented to an analog image processor for a high-speed edge image enhancement operation. The processed image data, together with the original raw image data, is fed into a digital image processor for further processing. In this step, the digital image processor does not need to process all the image data; instead, it will analyze processed image data from the image processor and determined local regions (e.g., image contour with a broken edge). The digital DSP processor will process only data from those local regions accurately.

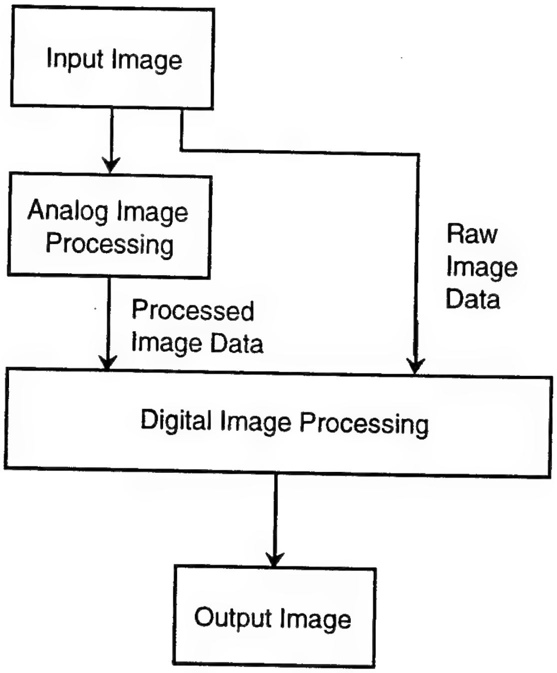


Figure 3-15
Flow chart of computer simulation.

A possible application of CNN in image processing is shown in Figure 3-16, in which a personal computer is used for data control and software-hardware co-design integration.

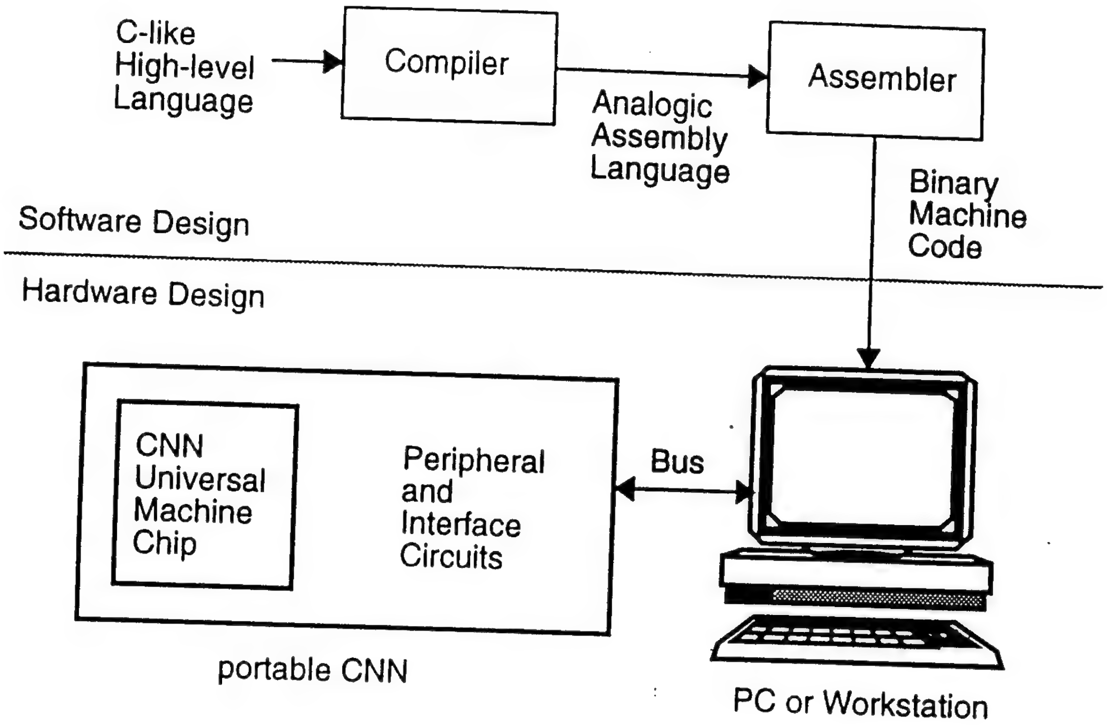


Figure 3-16
Software-hardware co-design scheme.

In this example, C-like high level language is translated into assembly language, which is further translated into binary machine code by the assembler after compiling. Then, the machine code is executed by the host PC. A CNN universal machine chip works as a coprocessor inside the host machine. The CNN universal machine board can also communicate with the host PC through a standard bus interface, such as a PCI, ISA bus.

In this system, two different clock signals are used, one for analog processing and the other for digital operation. The analog processing program and template data can be stored in built-in RAM or off-chip RAM. During the compilation stage, techniques such as pipeline can be applied. For example, by using a sequential loop, the host PC can load the next template data, executing a CNN instruction, while producing the previous results.

A sample C-like high-level language program is written as :

Program sample: Edge Detection

```
/* Declaration of  variable */
```

```
        AIMG         M1;   /*  gray-level image  */
        DIMG         M2, M3, M4, M5, M6; /* binary value images  */
/* AVERAGE, EDGE, VEDGE, DEDGE, LAND  are pre-defined functions. */

Begin
        INIT;
        LOAD(M1);
        M2 = AVERAGE(M1);
        M3 = EDGE(M2);
        M4 = VEDGE(M3);
        M5 = DEDGE(M3);
        M6 = LAND(M4, M5);
        OUT(M6);
end
```

The assembly code, which was translated from the above C-like program by the compiler, is shown as follows:

```
BEGIN       ;PROGRAM START
;LOAD TEMPLATE INFORMATION
SELAPR 0    ;LOAD TEMPLATE 0 INFORMATION TO APR 0
LDAPR 0
SELAPR 1    ;LOAD TEMPLATE 0 INFORMATION TO APR 1
LDAPR 1
SELAPR 2    ;LOAD TEMPLATE 0 INFORMATION TO APR 2
LDAPR 2
SELAPR 3    ;LOAD TEMPLATE 0 INFORMATION TO APR 3
LDAPR 3
RESET       ;RESET LOCAL NUCLEU
INPUT               ;INPUT SAMPLE AND HOLD
;GET AVERAGE BLACK AND WHITE IMAGE FROM GRAY SCALE IMAGE
TEMP 0      ;SELECT TEMPLATE FROM APR 0
CNN         CNN TRANSLATION
STO4 0      ;STORE THE OUTPUT IN LAM4(0)
FBACK 0     ;FEEDBACK LAM4(0) TO THE INPUT/INITIAL STATE
;GET EDGE-DETECTED IMAGE
```

```
TEMP 1      ;SELECT TEMPLATE FROM APR 1
CNN         CNN TRANSLATION
STO4 1      ;STORE THE OUTPUT IN LAM4(1)
FBACK 1     ;FEEDBACK LAM4(1) TO THE INPUT/INITIAL STATE
;DETECT VERTICAL EDGE
TEMP 2      ;SELECT TEMPLATE FROM APR 2
CNN         CNN TRANSLATION
STL 0       ;STORE LAOU TO LLM(0)
FBACK 1     ;FEEDBACK LAM4(1) TO THE INPUT/INITIAL STATE
;DETECT DIAGONAL EDGE
TEMP 3      ;SELECT TEMPLATE FROM APR 3
CNN         CNN TRANSLATION
STL 1       ;STORE LAOU TO LLM(1)
;AND OPERATION OF THE PREVIOUS TWO RESULTS
LLM 0       ;ACTIVATE THE LLM(0) TO THE INPUT OF LOGIC FUNCTION
LLM 1       ;ACTIVATE THE LLM(1) TO THE INPUT OF LOGIC FUNCTION
LDAND       ;LOAD THE AND FUNCTION TO THE LLU
LOUT                ;SEND THE LOGIC FUNCTION RESULT TO THE OUTPUT LINE
LDEA 0      ;DEACTIVATE LLM(0)
LDEA 1      ;DEACTIVATE LLM(1)
END        ;PROGRAM TERMINATE AND SENDS OUTPUT READY SIGNAL
```

Table 3-3 lists the assembly instructions with a brief explanation. Assembly language code is further translated into machine code to be uploaded to array processor hardware. The code generated from the above program looks like the following code, except the words after the '!' sign, which belong to a corresponding assembly program.

```
0110        0000        !BEGIN
1101        0000        !SELAPR 0
1110        0000        !LDAPR 0
1101        0001        !SELAPR 1
1110        0001        !LDAPR 1
1101        0002        !SELAPR 2
1110        0002        !LDAPR 2
1101        0003        !SELAPR 3
1110        0003        !LDAPR3
```

```
0000    0000    !RESET
0001    0000    !INPUT
0111    0000    !TEMP 0
0010    0000    !CNN
0011    0000    !STO4 1
0101    0000    !FBACK 0
0111    0001    !TEMP 1
0010    0000    !CNN
0011    0001    !STO4 1
0101    0001    !FBACK 1
0111    0002    !TEMP 2
0010    0000    !CNN
0100    0000    !STL 0
0101    0001    !FBACK 1
0111    0003    !TEMP 3
0010    0000    !CNN
0100    0001    !STL 1
1001    0000    !LLM 0
1001    0001    !LLM 1
1000    0000    !LDAND
1010    0000    !LOUT
1011    0000    !LDEA 0
1011    0001    !LDEA 1
1111    0000    !END
```

Table 3-3 Assembly Instruction Sets

Analogic Instruction		Mnemonic instruction	Explanation	Equivalent SCR Configuration
First 4 bits	Last 4 bits			
0000	0000	RESET	reset local nucleus	SEL(S0)
0001	0000	INPUT	input sample and hold	SEL(S1)
0010	0000	CNN	start CNN translation	SEL(S2)
0011	X	STO4	store the output from LAOU to LAM4(X)	SEL(S3)
0100	Y	STL	store the output from LAOU to LLL(Y)	SEL(S4)
0101	X	FBACK	feedback LAM4(X) to the input/initial state	SEL(S5)
0110	0000	BEGIN	program begin	BEGIN
0111	Z	TEMP	select template from APR(Z)	SEL(TEM)
1000	W	LAND,LOR LNOT	select the desired logic function	SEL(S6)
1001	Y	LLM	activate LLM(Y) to the input of the logic function	
1010	0000	LOUT	send the logic function result to the output line	
1011	Y	LDEA	deactivate LLM(Y) to the input of the logic function	
1100	0000		undefined	
1101	Z	SELAR	select APR(Z) to load the CNN template information	EXTERNAL SETUP
1110	0000	LDAPR	load template information to the activated APR	
1111	0000	END	program terminates and sends OUTPUT READY signal	END

3.6 CNN Development Toolkits

In addition to these CNN chips, several toolkits were also expected from Celnet. These toolkits can be described as follows.

3.6.1 The CNN Application Development Environment and Toolkit (CADET)

The CNN Application Development Environment and Toolkit (CADET) assist the application engineer with designing and testing analogic CNN algorithms and software in a real-life environment. The package features:

- Multilayer simulators for testing CNN instructions (templates) and sequences of instructions
- Learning programs for designing CNN templates
- High level analogic CNN language (ACL) and compiler
- Analogic CNN program library

- Application case studies
- Hardware accelerator add-in-board for solving problems up to 1 million pixels with 2 μs/cell/iteration speed
- Various optical interfaces and devices including video, CNN camera, and scanner interfaces
- Algorithms created in the ACL language can be used directly to test and prototype actual CNN processors on the accompanying CNN Chip Prototyping System (CCPS).

In addition to design tools, CADET provides a real-life test environment with various imaging input, output devices, and interfaces. The analogic CNN Program Library contains more than 100 different CNN instructions (templates), as well as many useful subroutines and programs. A special genetic learning algorithm is also available for assistance with template design.

The CADET system is hosted by an IBM compatible PC (386 or better CPU, VGA graphics, ISA bus, 4 MB RAM, hard disk), and has the following capablities:

CNNM Multilayer CNN Simulator

- Simulates multiple layers with possible different cell densities of continuous and discrete time CNNs
- Linear, nonlinear, and delay-type templates connecting arbitrary layers
- Template and image sizes limited only by memory capacity
- Easy-to-use menu-driven environment, operable by keyboard or mouse
- Hot keys, command line options, and files utilizing various configurations for experienced users
- Easily configurable display features
- Recording capabilities for transients.

ACL Language and Compiler

- C-like programming language and libraries
- Special commands for image acquisition and analogic processing
- Compiler-generated code for the CNNM simulator and the hardware accelerator
- ACL code directly usable for driving analog VLSI chipsets on the CCPS.

Template Designer

- CNN template learning program based on genetic optimization algorithms
- Supervised and unsupervised learning
- Support for implementing application specific learning constraints.

Hardware Accelerator Board (HAB)

- Four TI TMS320C25 DSP

- 8 MB on-board memory
- ISA card format
- 2 µs/cell/iteration processing speed
- 1 million cell (pixel) capacity.

Interfaces

- Most popular formats (BMP, TGA, PIC, BMP, etc.) are supported (also facilitates optical scanner input)
- Video signal (PAL, NTSC) input through frame grabber option
- CNN camera input
- C-mount microscopy input option
- Input option for some X-ray scanners.

Analogic CNN Program Library

- CNN templates for basic image processing, logical and morphological operations, color processing, texture classification, motion extraction, etc.
- CNN algorithms for object counting, scratch removal, textile defect detection, banknote recognition, finding the shortest path, etc.

3.6.2 The CNN Chip-Prototyping System (CCPS)

The analogic CNN chip-prototyping system was designed and built for two main purposes. The first is to test CNN Universal Chips (cellular Processor, cP) or chipsets, other programmable and fixed template CNN chips, and their subassemblies. After testing and verifying the chips, the other function of the system is to run analogic CNN algorithms, developed on the CNN Application Development Toolkit (CADET), directly on the CNN chip. This second function is supported by the same high-level language (Analogic CNN Language, or ACL) used in CADET.

Hence, fully tested cP-s with loaded algorithms will be inserted into boards designed for various applications that use this new technology. The following provides the capabilities of the CCPS:

Features

- Analogic CNN algorithms can be defined by a high-level language, ACL (Analogic CNN Language).
- An analogic CNN algorithm library, including the templates, is available.
- The system can be used without special programming practice.

- An intermediate assembler code, analogic CNN machine code (ACMC), can be used for directly programming the cP.
- A unique CNN physical interface is used for all CNN chips (CNN platform bus).
- Each type of cP has its own dedicated platform connected through the CNN Platform bus.
- Input sources: CCD imagery chips in a camera
  On-chip optical input with different optical interfaces
  Previously stored images.

Architecture

Binary control, as well as analog and logic data streams for the CNN chips, is communicated through the CNN Platform bus, incorporating:

- Template sequences
- Local logic operator sequences
- Switch configuration sequences
- Code for the global analogic control unit
- I/O information for the CNN array.

The CNN Chip Prototyping System is a three-level structure (Figure 3-17). The top level is a PC (host machine). It provides user interface and controls the CNN Prototyping System board (CPS board). The ACL program that runs on this level is independent of the currently tested/used CNN chip.

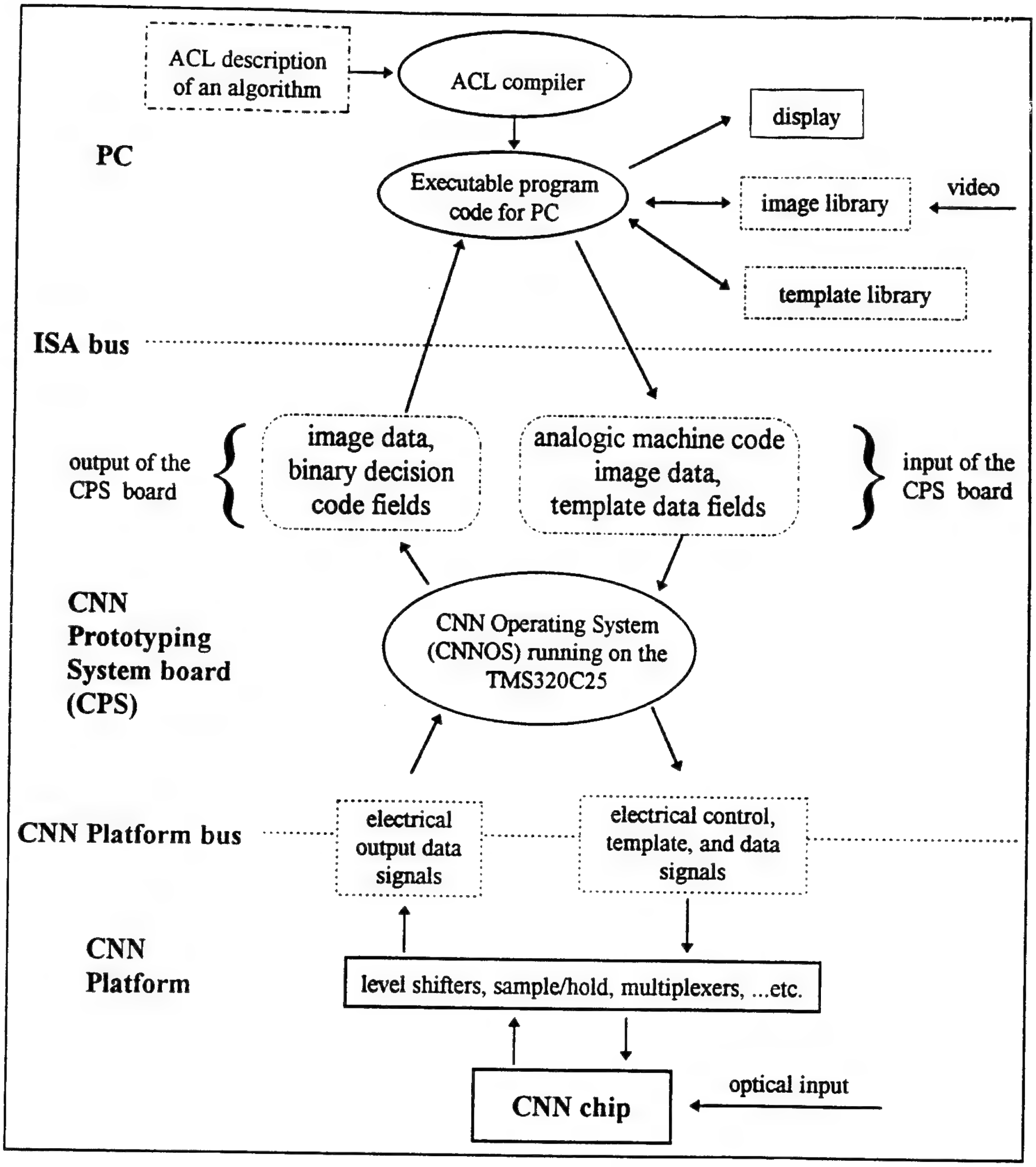

ACL description of an algorithm
ACL compiler
PC
Executable program code for PC
display
image library
video
template library
ISA bus
output of the CPS board
image data, binary decision code fields
analogic machine code image data, template data fields
input of the CPS board
CNN Prototyping System board (CPS)
CNN Operating System (CNNOS) running on the TMS320C25
CNN Platform bus
electrical output data signals
electrical control, template, and data signals
CNN Platform
level shifters, sample/hold, multiplexers, ...etc.
CNN chip
optical input

Figure 3-17
Operation of the CNN prototyping system.

The median level is the CPS board, which is controlled by the intermediate code (ACMC) generated by the ACL compiler. It controls the CNN Platform according to the currently tested/used CNN chip hardware specification, following the commands of the host machine. This level is driven by a TMS320C25 microprocessor. Its output is a standard CNN Physical Interface code (CPI code).

The bottom level is the CNN platform. It hosts the CNN chip. Its role is to adapt the CNN platform bus signals (CPI code) of the CPS board to the current CNN chip. This platform board performs TTL-CMOS-TTL conversion, serial-parallel-serial multiplexing and sample holding, analog level shifting, and control code decoding. Because of the different CNN chip designs, CNN Platforms are designed specifically for each new CNN chip type.

For CNN chips with on-chip optical input, an optical mount, coupled to the CNN platform, is provided.

The PC-CPS board communication is done through the standard ISA bus (AT bus). The CPS board-CNN platform communicates through the CNN platform bus (using ribbon and coax cables).

Software

To support analogic CNN algorithm development, a high-level, easy-to-use language was defined, and a new version of our compiler was developed (ACL V.2.1). Because the CNN array works together with a digital processor of the host machine in many applications, the ACL is implemented as a C library; thus, the functions of the digital processor and the CNN chip can be combined.

The compiled code of the ACL compiler is an executable program for the PC, which generates and downloads analogic CNN machine code and data (images) to the CPS board. The CPS board executes the algorithm (driving the platform and the cP chip), and the results and subresults are uploaded, displayed, and saved, according to the ACL program.

3.7 CNN Chip Fabrication

The CNN chips were designed and fabricated by Celnet, a California-based company. Several design efforts were made by Celnet during this program. Table 3-4 shows the chipset specification proposed by Celnet.

Table 3-4 CNNU Chip Specification Sheet

Processor:	cP1000	cP1000A	cP4000A
processor space:	32x32	32x32	64x64
neighborhood:	3x3	3x3	3x3
time constant:	200ns	200ns	200ns
weight accuracy:	2%	2%	2%
LAMs (2% < 100μs):	No	4	4
LLMs:	4	4+1	4+1
LLU I/O:	2/1	2/1	2/1
Local logic operation time:	100ns	100ns	100ns
image concurrent up/down loading:	No	Yes	Yes
Input			
levels:	TTL, binary	1V-4V analog	1V-4V analog
parallel:	32 channel, binary (10MHz)	32 channel analog (5MHz)	64 channel analog (5MHz)
serial:	No	analog (10 MHz)	analog (10 MHz)
optical:	YES (binary)	YES (analog)	YES (analog)
template loading time:	1.9μs	1.9μs	1.9μs
Output			
parallel:	32 channel, binary (10MHz)	32 channel analog (5MHz)	64 channel analog (5MHz)
digital flag (all black):	No	Yes	Yes
GAPU			
on-chip stored templates:	8	10	10
global interrupt:	No	Yes	Yes
External analog image storage chip set support			
name:	-	ARAM32-1	ARAM64-1
size:	-	32x32xN	64x64xN
storage time:	-	200ms (2%)	200ms (2%)
accuracy:	-	8 bit	8 bit
on-chip ADC/DAC:	-	4	8

The delivered system from Celnet includes a plug-in board, CNN chipset (for binary image), CNN development box, and control software. The plug-in board is shown in Figure 3-18. Its function is to interface the CNN development box with the computer. An ISA bus was used for this board. Processing data, control comment, and CNN template format are controlled through this ISA bus. Figure 3-19 is the delivered 32 × 32 cP1000 CNN chipset and the development box from Celnet. The function of this CNN chipset is to perform several image processing for binary image (black-and-white) by applying different templates. We examined the layout of this chipset carefully and found that it is a digital-based chip design (i.e., CNN circuitry was implemented in the digital domain, rather than in the analog domain). The development box contains the CNN operation and interface circuits. The CNN chipset is installed in this box and all CNN operations are performed in this box. Operation of this box is controlled by the plug-in board so control comment can be input from the computer. Several templates are associated with this system from Celnet. These templates are implemented with software coding. Figure 3-20 shows the menu for this software package. This main menu includes input image, input template, running the CNN, and saving the output image. Only binary images can be processed in this package. Several templates have been installed in this software, including edge detection, hole filling, image enhancement, and etc. Comments can be input from this menu, and the operation will be performed in the CNN box. Figure 3-21 shows the edge detection for a binary image, where left is the original image and right is the resulted image. In conclusion, the delivered system from Celnet includes an interface board, a CNN development box, and a software package to control the CNN operation. This system is intended for a binary only image, which is always pre-stored in the computer.

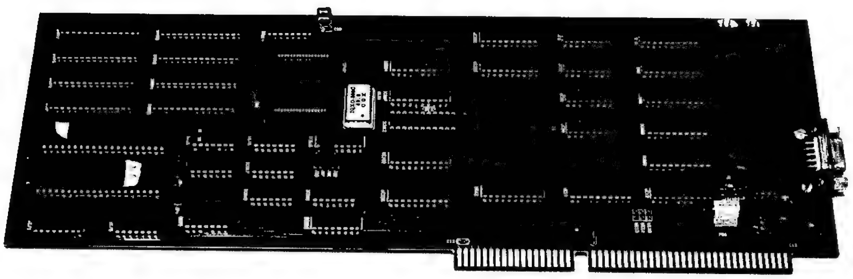


Figure 3-18
A CNN plug-in board for PC interface.

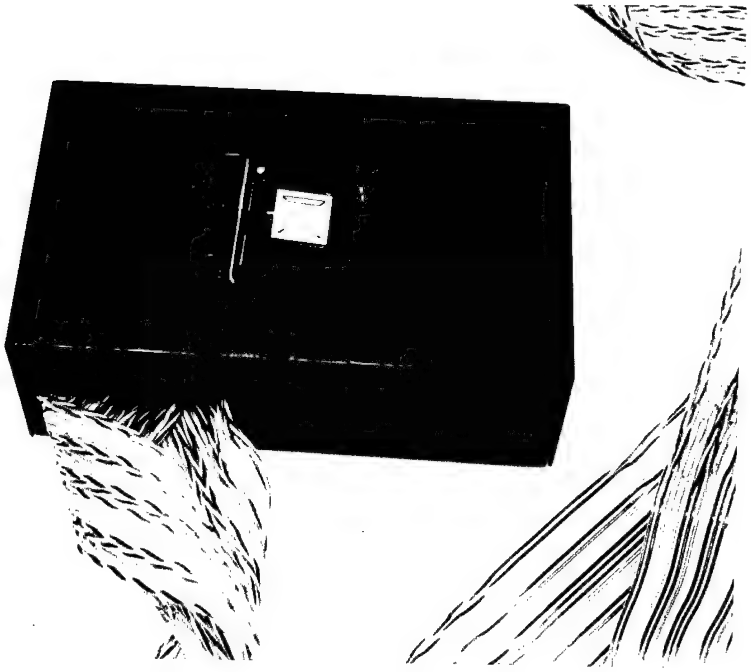
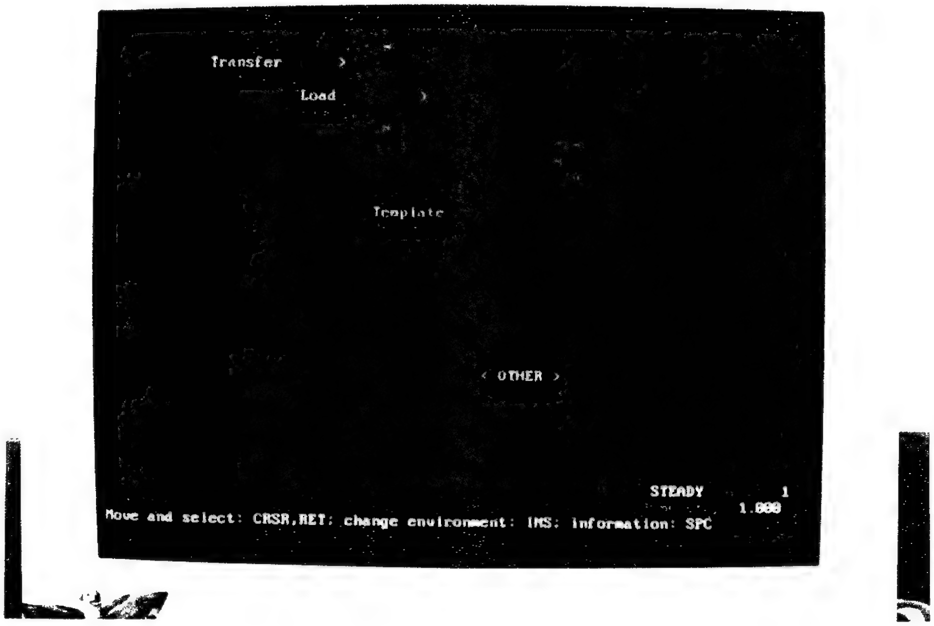
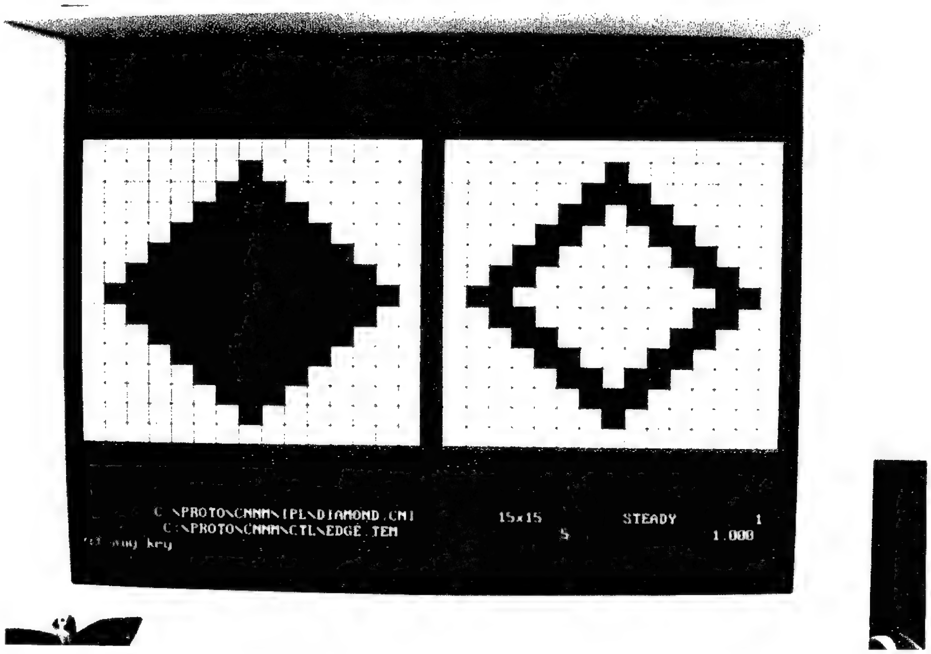


Figure 3-19
A CNN development box for evaluating and testing CNN chips.



```
                                                         STEADY        1
                                                                   1.000
Move and select: CRSR,RET: change environment: INS: information: SPC
```

Figure 3-20
A computer screen for CNN operation menu.



```
      C \PROTO\CNNM\IPL\DIAMOND.CNI          15x15        STEADY        1
       C:\PROTO\CNNM\CTL\EDGE.TEM                  5               1.000
[illegible] any key
```

Figure 3-21
The experimental results for edge detection.

Based on these chipset designs, Figure 3-22 shows system design of the cP4000A CNN chipset. In this system, the SRAM, analog RAM, and 64-bit microprocessor are incorporated with the CNN chip. This system can process each frame in 3 microseconds, and output each frame in 12.8 microseconds. With this speed, real-time image processing can be achieved. Figure 3-23 shows another CNN image processing system implementation with a more powerful CNN chip (100 × 100 cells). In addition to a more powerful CNN chip, the difference between this system and Figure 3-22 is the use of a sample-and-hold D/A converter instead of analog RAM. Table 3-5 shows speed comparison between these two different implementations. A comparison of total operating time between these two systems is shown in Figure 3-24. From this figure, we find that the 64 × 64 CNNUM with A-RAM has better performance in the average operation range (5 to 35) than the 100 × 100 CNNUM with sample-and-hold.

Table 3-5 Comparison of 64 x 64 CNN Chip with A-RAM and the 100 x 100 CNN Chip

	64 x 64 CNN with A-RAM	*100 x 100 CNN*
loading time	3 ns/pixel	20 ns/pixel
operation time	0.75 ns/pixel/op.	0.3 ns/pixel/op.

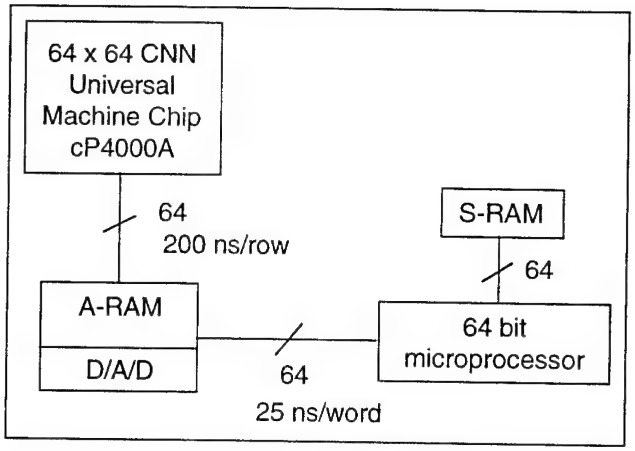


Figure 3-22
Image processing unit based on 64 x 64 CNN Universal Machine Chip (cP4000A) and analog RAM.

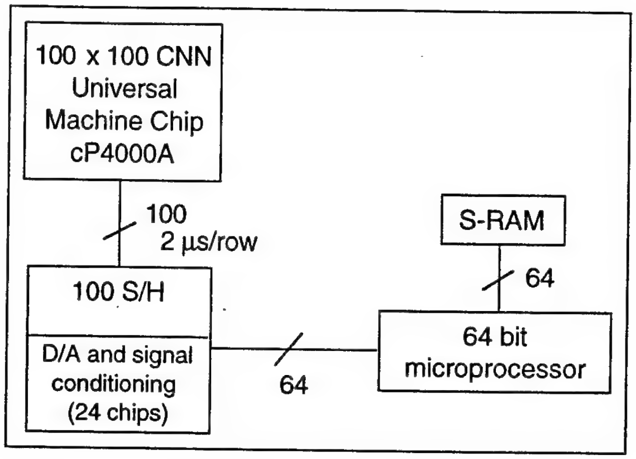

100 x 100 CNN Universal Machine Chip cP4000A
100
2 µs/row
100 S/H
D/A and signal conditioning (24 chips)
64
S-RAM
64
64 bit microprocessor

Figure 3-23
Image processing unit based on 100 x 100 CNN Universal Machine Chip.

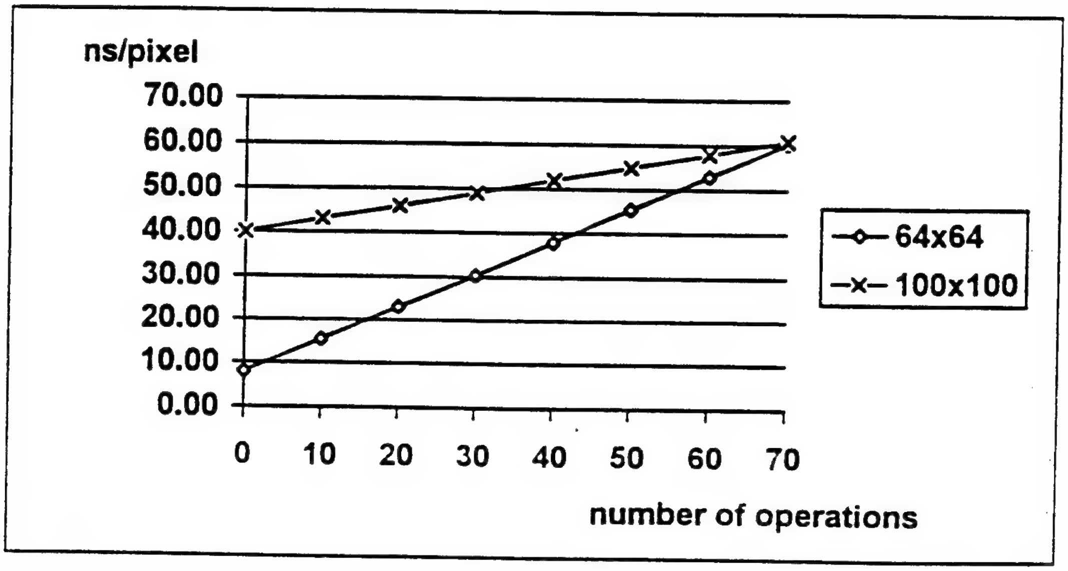


Figure 3-24
The average number of operation is from 5 to 35. In this region the 64 x 64 CNN chip width A-RAM is faster than the 100 x 100 CNN chip.

Because of the special characteristic of the video signal (NTSC video output standard is used), several efforts must be implemented to interface the CNN system with a video signal. Figure 3-25 shows the design of an interface system for the CNN universal machine with a video camera. In the NTSC standard, the video signal is in serial format. The image is scanned from top left to down right. In addition, an interlacing signal is always used in the NTSC standard. This means that the video signal is output with odd lines and even separate lines. To integrate this serial-interlacing signal with the parallel-processing CNN modulo, we must use buffering technology. In this system design, an A/D converter and a memory device are used to make the serial NTSC signal compatible with the parallel characteristics of the CNN machine. As shown in Figure 3-25, the analog video signal is processed and converted into digital format. The function of signal processing is to extract useful information from the NTSC format, since it has some overhead and redundant signals. After that, digital signals are stored in the buffer for further processing. The memory buffer stores serial digital information and outputs them in parallel format. These parallel signals are then converted into an analog signal, which can be interfaced easily with the CNN universal machine. With this design, the video signal can be synchronized and integrated in a format suitable for CNN processing.

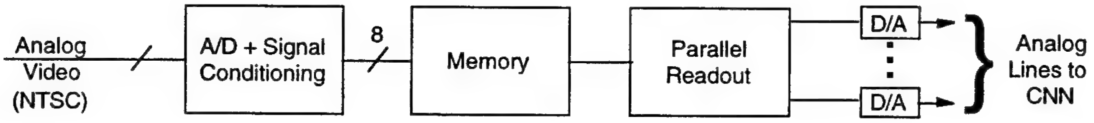


Figure 3-25
CNN interface with NTSC signal.

The entire CNN chip interface solution is shown in Figure 3-26. This design is based on 32 × 32 CNN chipset. A larger throughput system can be implemented easily with the same method by expanding the data throughput. In this system, the NTSC signal from video camera is amplified first by an AGC (automatic gain controller). After that, a DC restore and A/D converter (8-bit) are used to convert the NTSC signal into a digital signal, which describes the image frame in digital format. These digital data are stored in the dual port V-RAM. Address generators (both writing and reading) are incorporated with the V-RAM to manage the address mapping and retrieval. This address information is applied to the data after CNN processing, so these processed data can be reconstructed into a useful video format. A 32 × 8 serial/parallel buffer is used to convert serial data into parallel data. Since the data format is in digital form, we need to further convert the data into analog format for CNN processing. After CNN processing, the processed data, combined with address information, is input into circuitry that performs pixel combination and display synchronization. With this circuitry, the processed data can be formatted compatibly with the NTSC signal.

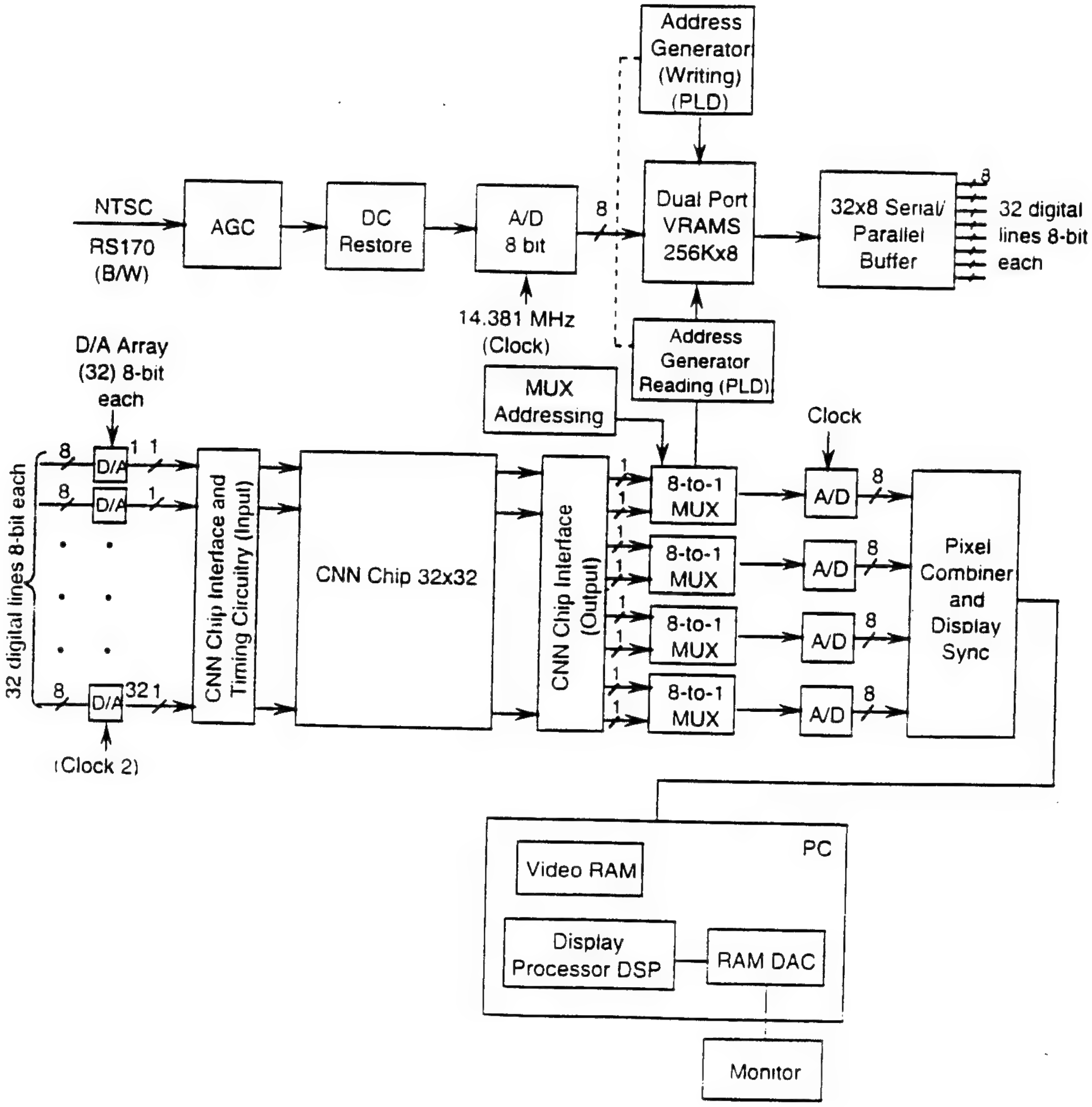

Address Generator (Writing) (PLD)
NTSC
RS170 (B/W)
AGC
DC Restore
A/D 8 bit
8
Dual Port VRAMS 256Kx8
32x8 Serial/ Parallel Buffer
8
32 digital lines 8-bit each
14.381 MHz (Clock)
Address Generator Reading (PLD)
D/A Array (32) 8-bit each
MUX Addressing
Clock
32 digital lines 8-bit each
8
D/A
1 1
8
D/A
1
8
D/A
32 1
(Clock 2)
CNN Chip Interface and Timing Circuitry (Input)
CNN Chip 32x32
CNN Chip Interface (Output)
1
8-to-1 MUX
A/D
8
Pixel Combiner and Display Sync
PC
Video RAM
Display Processor DSP
RAM DAC
Monitor

Figure 3-26
CNN ship interface total solution.

4.0 DISCRETE-COMPONENT CNN SYSTEM DEMONSTRATION

Analog VLSI image processing techniques, such as silicon retinas, cellular neural networks or early-vision neural chips, are most suitable for front-end, early-vision processing (including edge enhancement and feature extraction) because of their advantages in processing throughput, chip area, low-power consumption, and cost.

The two principal drawbacks of analog VLSI imaging techniques are their lack of flexibility and their inaccuracy. These techniques are difficult to change after implementation. In the mean time, because of the inherent property of the MOS transistor, the accuracy of these techniques is low. In particular, the precision problem may become serious when the images are noisy and of low contrast. It is difficult to attain while processing low-contrast and high-clutter images. Pixels with higher or lower intensity tend to be images with low contrast or images in a highly cluttered background. This makes the analog VLSI processor unsuitable for images with low contrast or images in highly cluttered backgrounds.

In this analog circuit, both current and voltage must be controlled exactly. Unlike a digital circuit, the tolerance allowance for current and voltage in an analog circuit is very limited. This limitation makes VLSI implementation extremely difficult. Therefore, instead of developing a VLSI chip, POC implemented the CNN circuit with discrete circuit components. The advantages of discrete components include:

- Ease of implementation: Implementing CNN with discrete components is not as difficult as VLSI.
- Flexibility in design system to accommodate different scales.
- Ease in system debugging.

4.1 System Application Design

This section describes a system application based on a Cellular Neural Network (CNN) concept. Figure 4-1 shows a block diagram of the application. A CNN circuit module accepts a serial analog video stream (such as NTSC, PAL, or SECAM) and performs intelligent, programmable, and ultra-fast low level image processing based on the principles of the CNN universal machine. Because of the parallelism and speed of the CNN circuit, the incoming analog video stream can be processed in less than a fraction of a frame time (i.e., ~ 33 ms). The resulting signal will be a processed serial analog video stream in NTSC, PAL, or SECAM format. In this way, both unprocessed and processed video streams can be sent to an imaging frame grabber and computing platform for video image digitization and high level digital image processing.

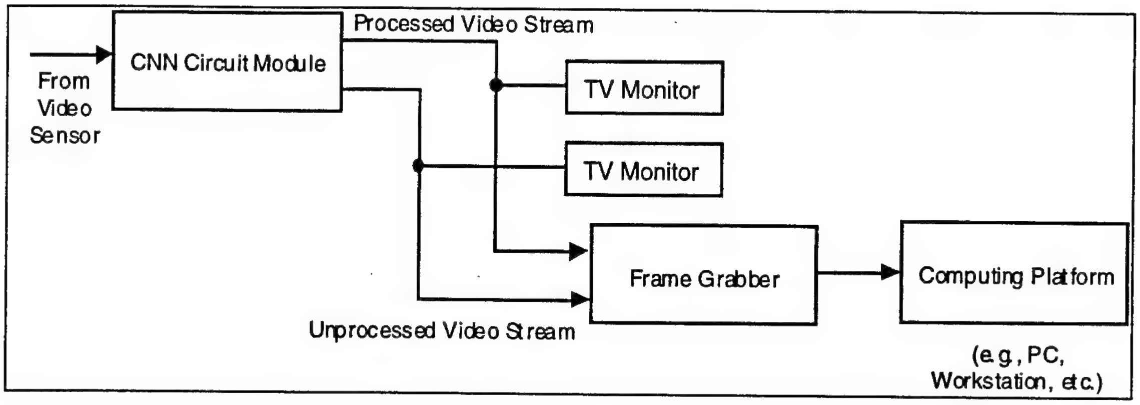


Figure 4-1
Block diagram of the CNN-based application.

Figure 4-2 illustrates the design of the CNN circuit module. A small set of CNN cells (e.g., 3×3, 5×5, 7×7, etc.) with a nearest local neighbor interconnect scheme is the heart of the circuit. Each CNN cell is equipped with functional circuits described in the CNN universal machine [2]. Delay line circuits are used to replicate the incoming video stream. The replicated video streams are delayed from one another by one horizontal video line. Each delayed video stream is input into a single row of the CNN cell array. In this way, CNN operation can be performed for the central CNN cell. The output is drawn only from the output of the central CNN cell. This output is then converted to a standard serial analog video format (NTSC, PAL, or SECAM); this is the processed video stream. The unprocessed video stream can also be easily drawn from the CNN circuit module, and is synchronized with the processed video system. The following summarizes the features of this system design:

1. Application architecture is novel because it provides synchronized unprocessed and processed serial analog video streams to an image video digitalization module (e.g., frame grabber) and/or a computing platform.

2. The CNN circuit module can be integrated with a standard video sensor (e.g., a CCD camera) as an intelligent front-end sensor, or with an image frame grabber as an integrated part of the computing platform.

3. The CNN circuit module receives a standard serial video format (NTSC, PAL, or SECAM) and outputs the same standard serial video format to widen its applications with standard imaging and video related equipment.

4. Only a small array of CNN cells (3×3, 5×5, or 7×7) is needed to process a large video image frame (e.g., 640×480) in real-time.

5. The incoming video stream is partitioned into several delayed video streams for CNN operation. This maximizes the use of time and spatial domains for real-time image or video image processing.

6. The use of processed video image data can provide faster than real-time video image pre-processing (such as edge enhancement, edge detection, thresholding, deblurring, half tone, etc.) before the data reaches the frame grabber or computing platform. Alternatively, it can provide the frame grabber/computing platform with pre-processed video/image information to reduce time for digital image processing by examining only critical regions (i.e., regions that require further analysis).

This CNN system application can be used in almost every field of image of video image processing that traditionally uses unprocessed video/image data as the only input. Use of the disclosed scheme offers many advantages in speed, size, power, and cost of an image or video image processing system.

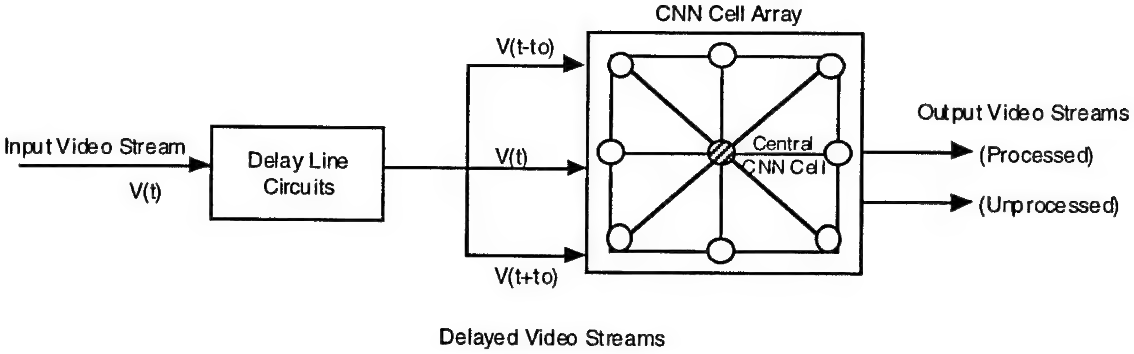


Figure 4-2
Design of the CNN circuit module.

4.2 Discrete Component-Based CNN Implementation

The circuit is designed to perform CNN operations on complete fields of television images in real time. It uses a mixture of digital and analog circuits, exploiting the strengths of each technique and minimizing the effects of their weaknesses. Digital circuits are far superior to analog circuits for storing information (instead of the A-RAM described in Section 3.0). A static RAM chip can store information indefinitely, with no drift or decay. This is difficult, if not impossible, with analog techniques. On the other hand, some operations, such as weighted sums, can be done by a few resistors at frequencies of several hundred megahertz if only moderate accuracy is required. A similar operation performed digitally could require hundreds of individual logic gates and operate at only a fraction of the speed. Our design was done as an exploratory breadboard. It is designed to operate at moderate speed with standard, easily obtainable parts. This design is shown in Figure 4-3 (a), (b), and (c). In the following, we will describe our design in detail.

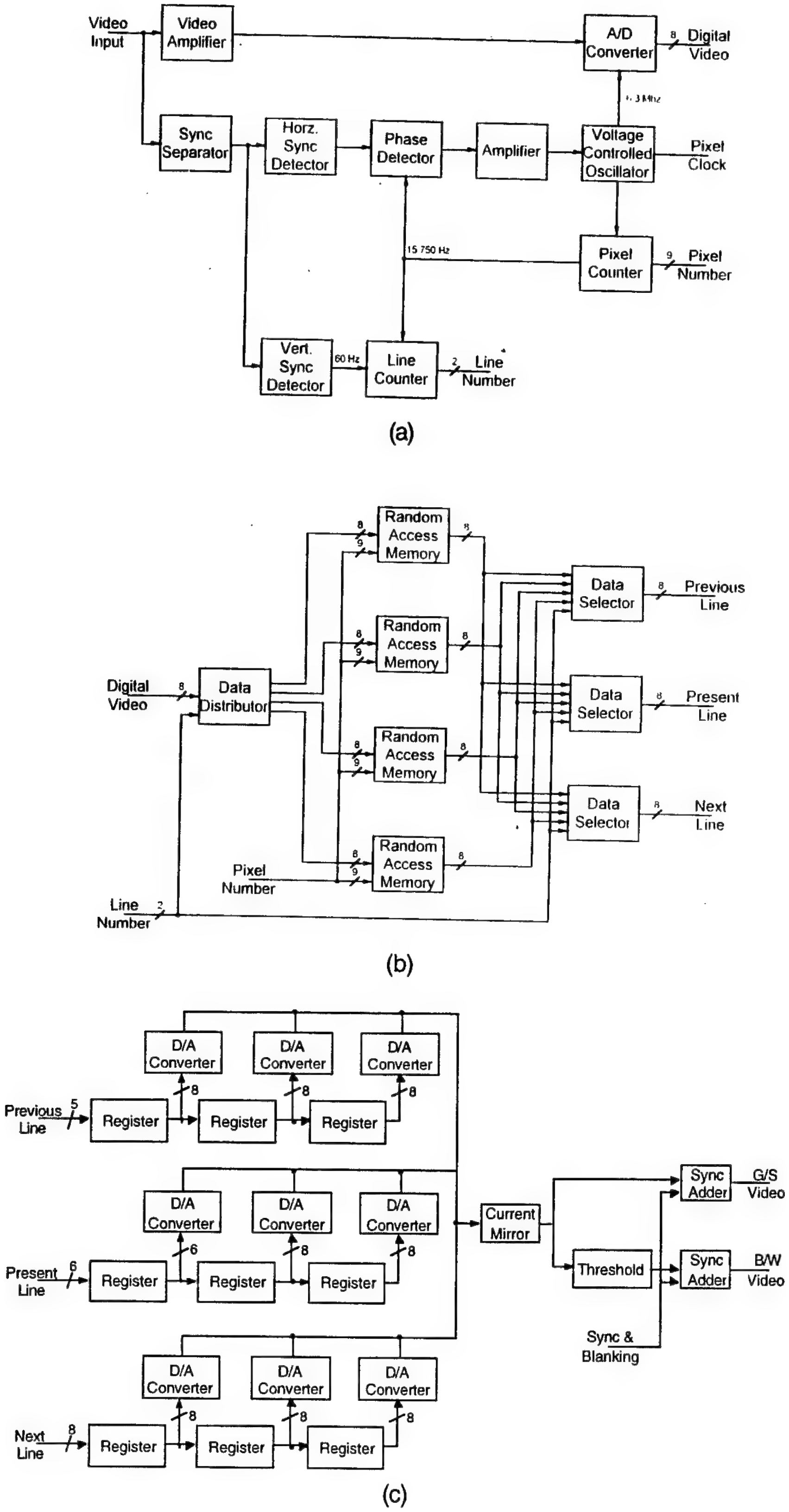


Figure 4-3
Hybrid video processor.

The processor performs operations on a single pixel and eight adjacent pixels at any given time. Processor output is a new value for a pixel that has been influenced by its spatial neighbors. This is done serially in real time at television rates. In raster-scanned video, only one pixel is available at a given time. The pixel to the left has already gone by, and the pixel to the right is the next one available. The top three neighbors were part of the previous line, and the bottom three neighbors will not be available until the next line is scanned. In this processor, three successive lines are stored digitally and read out in parallel for processing. Shift registers are used to store three adjacent pixels in each line, so at any given time the digital values of nine pixels are available. Nine multiplying D/A converters are used to convert digital values back to analog form. The multiplier on each converter represents a weighting coefficient for that pixel. The nine analog signals are summed together to form new video output, and sync and blanking pulses are added to allow viewing with a standard video monitor.

The processor is designed to work with a standard black and white video signal. First, synchronization information is extracted from the video signal with a sync separator. A monostable multivibrator with a period of three-quarters of a line is used to suppress additional equalizing pulses in the composite sync signal. The multivibrator output is used as the reference frequency for a phase-locked loop. This loop multiplies the horizontal frequency of 15,750 Hertz by 400 to generate a pixel clock of 6.3 Megahertz. A divider within the phase-locked loop is used to identify each pixel within a horizontal line and provides the pixel number to the random access memories. The 6.3 MHz pixel clock is used to trigger the A/D converter used to digitize the input video.

The output of the phase-locked loop divider is divided further by a line counter. This is used to provide blanking signals at the top and the bottom of the screen. The least significant two bits of the line counter are used to control the read and write sequence of the four random access memories. The vertical sync pulse is isolated from the composite sync signal by a monostable multivibrator with a period of one-quarter of a horizontal line, and is used to start the line counter during each field. The input video is amplified and digitized by an 8-bit A/D converter. The output of this converter is stored in written form in one of four memories, depending on the two least significant bits of the line counter. The storage address is determined by the nine least significant bits of the pixel counter. At any given time, only one memory is being written. The other three are read in a rotating sequence to three output buses. Both reads and writes are done to the same memory addresses. While one horizontal line is being written, the three previous lines are being recalled for processing.

These three data streams are each fed to three cascaded registers. Thus, at any given time, the registers contain data for three adjacent pixels on each of three adjacent lines. These nine registers provide the digital input for nine multiplying D/A converters. The full-scale value of each D/A converter can be set independently by changing the resistors in their reference circuit. This corresponds to a weighting factor for each pixel. The current outputs of the nine D/A converters are summed together, passed through a current mirror, and converted back to a voltage signal. This signal is split into two paths. One has composite sync and blanking added to it, and is output as gray scale video. The other is sent to a threshold competitor, which converts it to a one-bit digital signal; this signal has composite sync and blanking added before being output as a black and white signal. The processor is built in breadboard form on two boards. The interface between the boards is the three 8-bit buses feeding the register array. This resulted in comparable circuit density on the two boards, with 24 signals in the interface. Partitioning after the register array would have required 72 signals in the backplane.

The prototype boards used are in the 6U x 160 mm Eurocard format. These boards are equipped with two 96 pin connectors, which supply an adequate number of pins for digital buses and grounds. Card cases, cabinets, and power supplies are available from several vendors in this format, which will allow construction of a self-contained unit, if desired.

4.3 Future Expansion

Because of the short time available, this processor was built in a minimum configuration. It can only perform one operation on each pixel, and that is set by resistor values. These resistors can be replaced by low speed D/A converters, such as octal devices made by Analog Devices. This would require a digital control interface, possibly RS-232 compatible, to allow a laptop computer to be used as a controller. The interface between boards would permit other processor boards to be added in a daisy-chain fashion. If a sample-and-hold circuit were added to each board, this would allow the performance of several operations on each pixel in a pipe-line fashion at a real-time rate. Boards can be built for other operations, such as multiplying and squaring.

4.4 Design Details

Appendix A includes the detailed design of the CNN prototype board. The video input is designed to accept a standard RS-170 video signal. Input impedance is 75 ohms. This input is split into two paths. The sync signals are removed and used to generate timing signals for the video processor. The video portion of the signal is digitized, stored in line memories, and reformatted for video processing circuits. The video signal is first amplified by a factor of 2.47 in a high-speed op amp, and an offset of 1.65 volts is added. This results in a signal ranging between 1.55 and 3.25 volts. This matches the input range of the A/D converter. The A/D converter used is a TDA8703 manufactured by Phillips. It was used because of its availability. Other converters can be easily substituted if desired. The sample rate of conversion is 6.3 MHz. The output of the A/D converter is an 8-bit binary word. The data distributor consists of four 74HC244 octal buffers with inputs tied in parallel to eight data lines from the A/D converter. Each group of eight buffers feeds a separate random access memory (RAM). At any given time, only one set of buffers has its outputs enabled. This distributes the video to one of four RAMs. The RAMs are 8 K × 8 static memories. Each memory is driven by one 74HC244 octal buffer. When a buffer is enabled, its output is written into its associated memory. At any given time, one memory is being written and three memories are being read. Only 400 bytes of each memory are used. All the memories use the same address, which is generated by the pixel counter. The last two bits of the line counter are used to select the memory configuration. Thus, one memory stores the incoming video while the other three memories output the three previous lines. The data selector consists of three groups of four 74HC253 dual 4 to 1 multiplexers. Each group selects the output of one of the RAMs. The first group selects the memory that was written to three lines previously. The second group selects the memory that was written to two lines previously. The last group selects the previous line. The outputs of the data selector are three 8-bit signals that represent a vertical array of three pixels in the image. The data selector outputs are sent through the backplane to the next board. On the second board, three video signals are sent to a register array, which consists of nine 74HC374 octal latches. These are arranged as three groups consisting of eight three-stage shift registers. The contents of the registers consist of the last three pixels in each row of the three pixel column previously formed. This forms a square 3 × 3 array of pixels, which moves through the image at the 6.3 MHz rate set by the pixel clock. Each of these pixels is converted back to analog form by one of nine DAC0802 multiplying D/A converters. The weight of each pixel is determined by the reference input to each A/D converter. Converter outputs are in a current form and are summed to two common lines, one representing positive signals and the other representing negative signals. These two signals are fed to a current mirror, which subtracts the negative current from the positive current and transfers the difference to a load resistor to give a voltage output. A series diode is used to guarantee that output voltage cannot go below ground. A LM360 comparator is used to convert the analog video to a 1-bit digital signal by comparing it to a preset threshold.

The analog video goes to one of three paralleled emitter followers using PNP transistors sharing a common load resistor. The other two emitter followers are fed by blanking and sync signals. The most negative of the three signals determines the output level. During active field time, the blanking and sync signals are held above the white level, and the video signal determines the output level. During the blanking interval, the blanking signal is brought down to ground, and overrides the video signal, since it cannot go below ground. During sync pulses, the sync signal goes below ground and overrides both the video and blanking signals. The composite video signal produced is offset above ground by the base-emitter voltage of the PNP transistors. This offset is compensated by an emitter follower using an NPN transistor. This second emitter follower drives the output load through a serial 75-ohm resistor to provide reverse termination for the output cable. The output of the comparator is passed through a 74HC04 inverter to standardize its level, and then reduced to the white level of the video signal by a resistive divider. The result is then buffered and has blanking and sync pulses added to it in the same manner as the video signal. The video input feeds an emitter follower as well as the op amp in the main video signal path. The emitter follower drives a sync separator, which separates the composite sync from the video signal. The composite triggers two monostable multivibrators, which are the two sections of an MC74HC4328. One section is connected as a non-retriggerable multivibrator with a pulse width of about three-quarters of a horizontal line. This is triggered by each of the horizontal sync pulses. The extra equalizing pulses in the composite sync signal fall half a line after the horizontal sync pulses and fail to trigger the multivibrator since they fall within a pulse previously triggered by a horizontal sync pulse. The result is a regular pulse train at the horizontal line rate without extra pulses.

An MC4044 Phase Detector is used to compare the phase of this pulse train with the final stage of the pixel counter. The phase detector output is passed through a compensation amplifier and used to control the frequency of a voltage controlled oscillator. The output of the voltage controlled oscillator is at 6.3 MHz and is used as the pixel clock. The pixel clock is used directed by the A/D converter and is divided by a factor of 400 by the pixel counter. The output of the pixel counter is compared to the horizontal sync pulses in the MC4044 phase detector. The pixel counter is composed of three 74HC163 hexadecimal counters. The least significant nine bits of the counter are used as the address for the random access memories.

The second monostable multivibrator is set for a pulse width of one quarter of a horizontal line. The composite sync signal is sampled at the end of this pulse by a 74HC74 type D Flip-Flop. Sampling at this point determines the presence of the vertical sync pulse. The vertical sync pulse is used to start the line counter. The line counter consists of three 74HC163 hexadecimal counters. It is clocked by the line rate pulses from the pixel counter and started by the vertical sync pulse.

After a suitable delay, it unblanks the video and counts 256 lines. After 256 lines, it stops until it is reset by the next vertical sync pulse. The least significant two bits are used to determine the configuration of the data distribution and data selector in the video path. The stripped composite sync and blanking signals are also passed to the output amplifiers, where they are recombined with video outputs to form composite video signals.

4.5 Demonstration System and Experimental Results

Figure 1-1 shows the entire developed demonstration system, and Figure 4-4 shows the board that performs 3 × 3 cellular cells operation in real-time. In this system, the image is shot with a video camera and then input to the CNN board. The template of the CNN can be changed by changing resistor values. Several templates have been tested. Figure 1-2 (a) shows the original image and Figure 1-2 (b) shows the output image with an edge detection template. Figure 4-5 is another image scene with an edge detection template. An edge enhancement template was also tested; results are shown in Figure 4-6. Unfortunately, the photo in Figure 4-6 degrades the contrast of the edge and the background. It does not show good edge enhancement with the photo or with the naked eye.

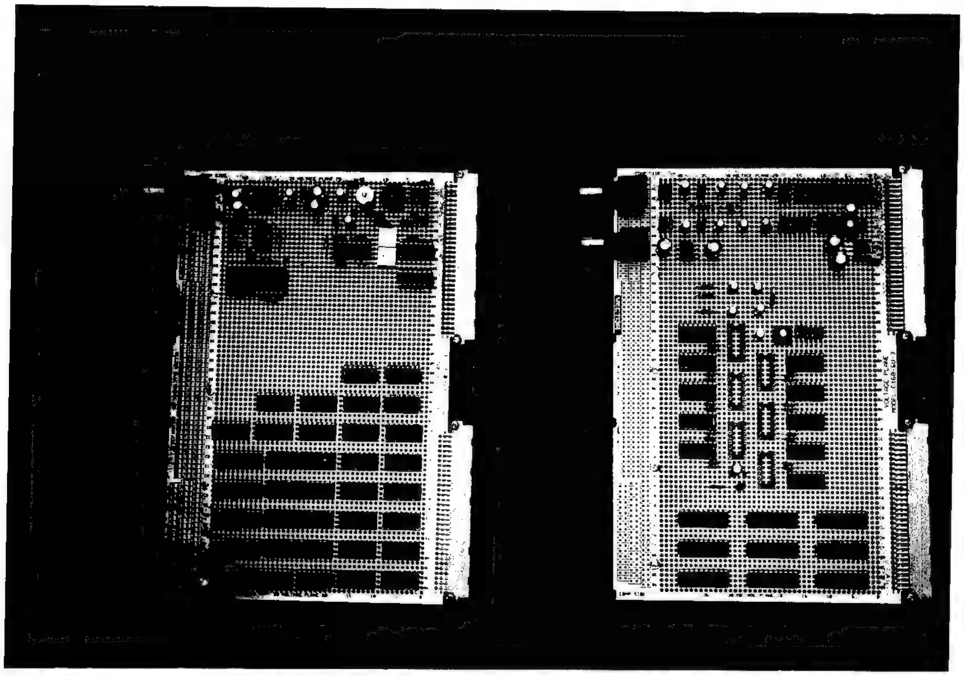
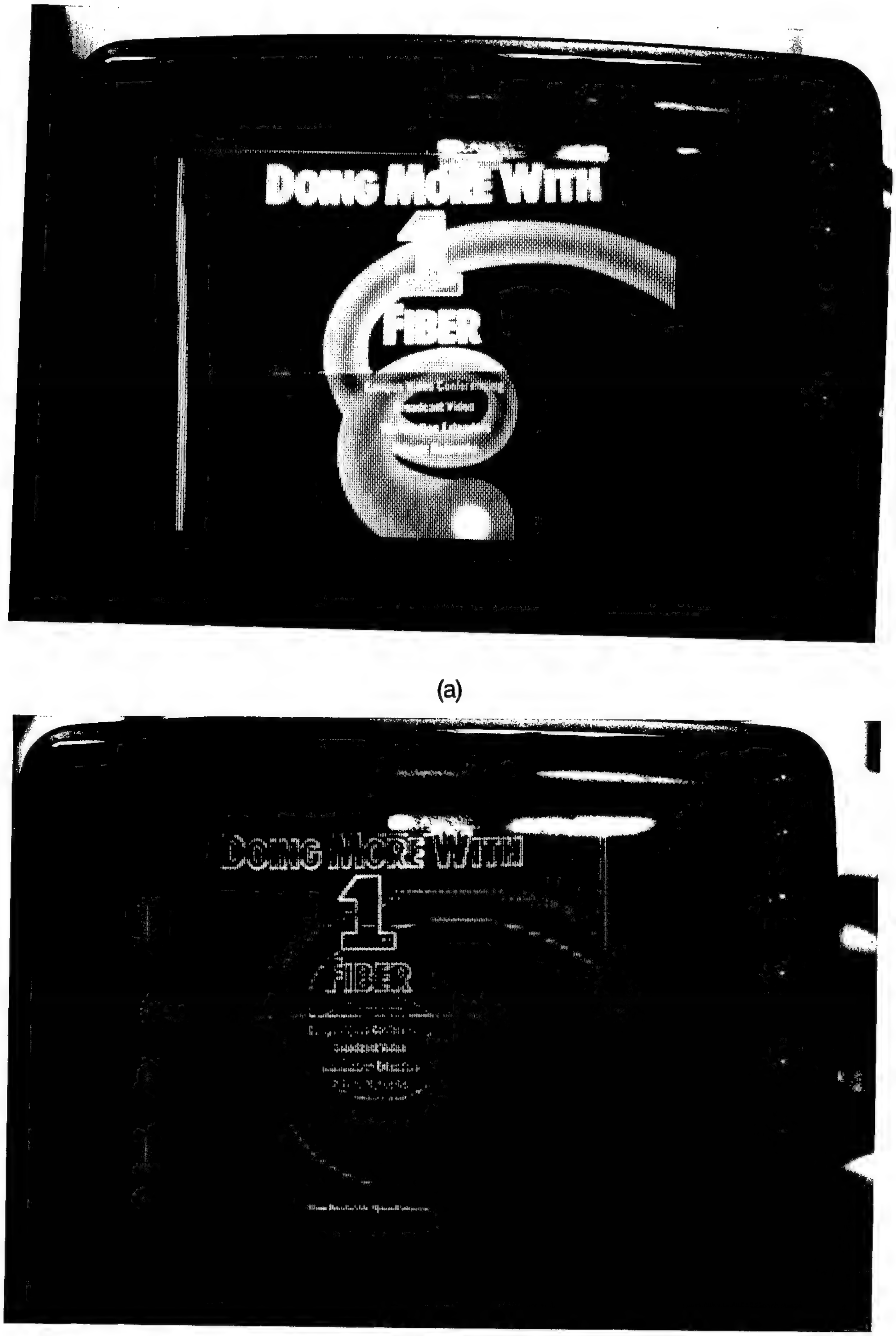


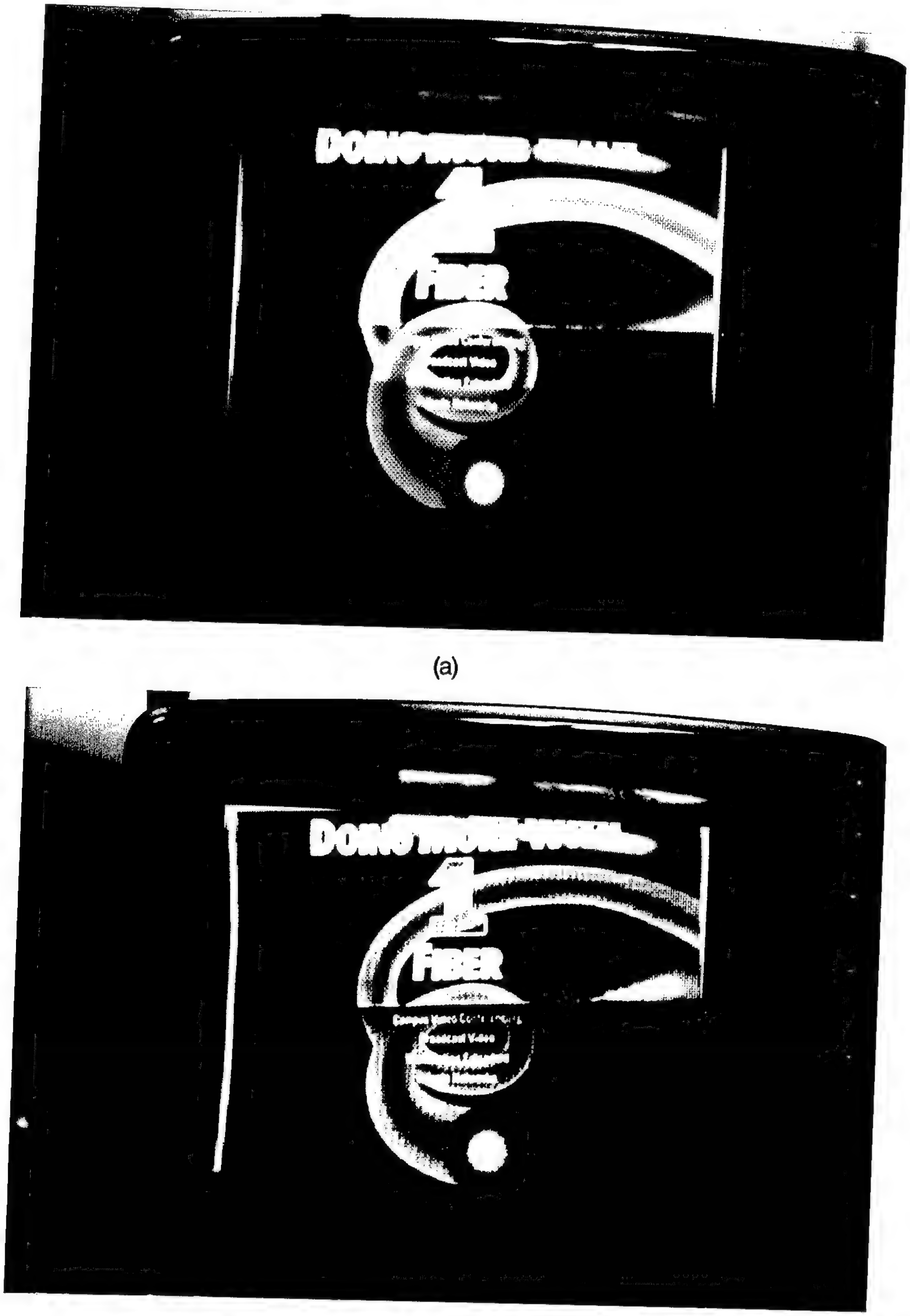
Figure 4-4
Two printed circuit boards designed for CNN system demonstration.



(a)

(b)

Figure 4-5
Edge detection operation: (a) original video image; (b) edge detected video image.



(a)

(b)

Figure 4-6
Edge enhancement operation: (a) original video image; (b) edge enhanced video image.

5.0 THE PHOTONIC INTERCONNECT AND ITS INTERFACE WITH THE DIGITAL PROCESSOR

This section will describe the other two elements, a photonic interconnect and a digital image processor, described in the system architecture of Section 2.0.

5.1 Basic Concept and Functionality of Photonic Interconnects

The advantages of using optical interconnect techniques rather than conventional electrical interconnect techniques are numerous. Besides the inherent advantages of optics, such as reduced electromagnetic and radio frequency interference and crosstalk, optical interconnections are capable of providing larger fanouts at higher bandwidths, with the possibility of lower system power and complexity. In addition, optics have an advantage over conventional interconnections in the density potential of free-space interconnections. Generally speaking, the advantages of optical techniques over conventional electrical approaches for control signal and data transfer are (1) wide bandwidth, (2) immunity to electrical interference and ground loops, and (3) long distance and high-speed capability. These characteristics simplify system design and hardware implementation, and permit more reasonable interconnect and control configurations than are possible or practicable with electrical approaches. The most dramatic advantage of optical communication is its capacity to easily handle high density interconnects at high data rates for both on-chip and off-chip applications.

In this program, a single chip implementation of the photonic interface function was proposed. Figure 5-1 shows a block diagram of an optical transceiver design. In this optical transceiver chip, a 32-bit data bus is used to communicate with the host machine, while one 8-bit local bus is used for optical receiving channels and another 8-bit data bus for optical transmitting channels. The chip consists of an I/O interface, a multiplexer (MUX), a demultiplexer (DEMUX), an optical transmitter (TX), and an optical receiver (RX). The CPU block represents the digital host machine. Each 32-bit data word from the host machine is segmented into 4 bytes and every byte is sent to the transmitter section for transmission through the laser diodes. At the receiver end, incoming optical signals are detected by the p-i-n diode array and demodulated into a signal by receiver circuits. Four bytes of data are combined through multiplexing circuitry to form a 32-bit word and then sent to the digital host machine. A detailed schematic of the I/O interface is given in Figure 5-2. Figure 5-3 shows a typical block diagram of the receiver circuit. It consists of pre-amplifier, automatic gain control amplifier, decision circuit, and clock recovery blocks. The clock recovery circuit can be omitted for short-distance communication. Incoming optical signals are detected and converted into electrical current signals by photodetectors, such as the p-i-n diode. The electrical current signals are amplified by the amplifier chain, which includes a low-noise pre-amplifier and an automatic gain control main amplifier. The decision circuit samples the signal and provides binary outputs through the thresholding function. The clock recovery circuit extracts the clock, which is used for sampling and further operation.

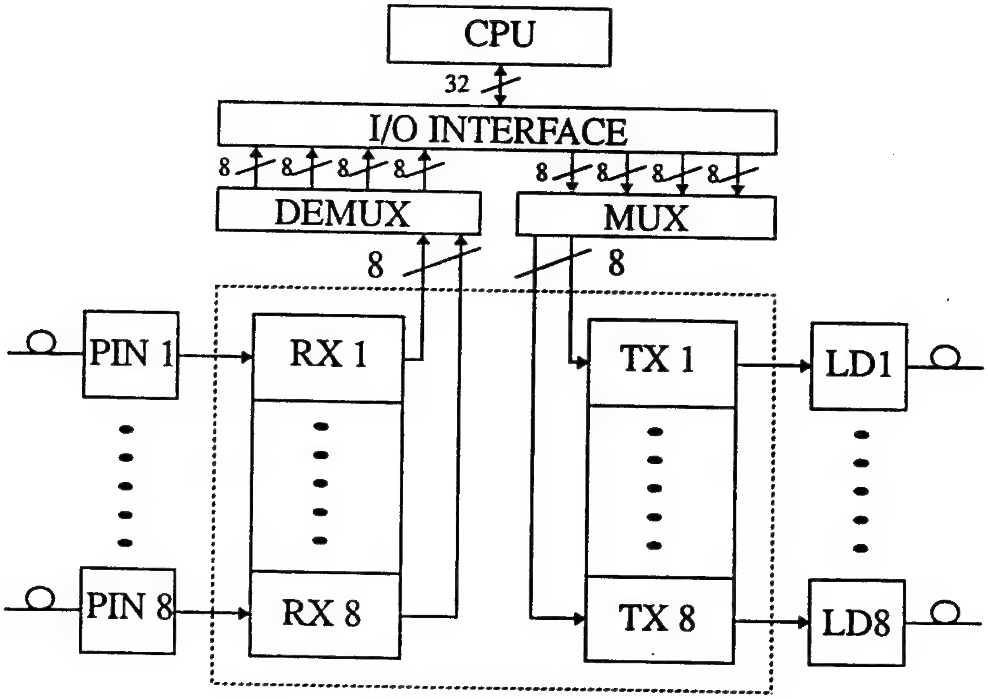


Figure 5-1
Block diagram of the optical interconnection system.

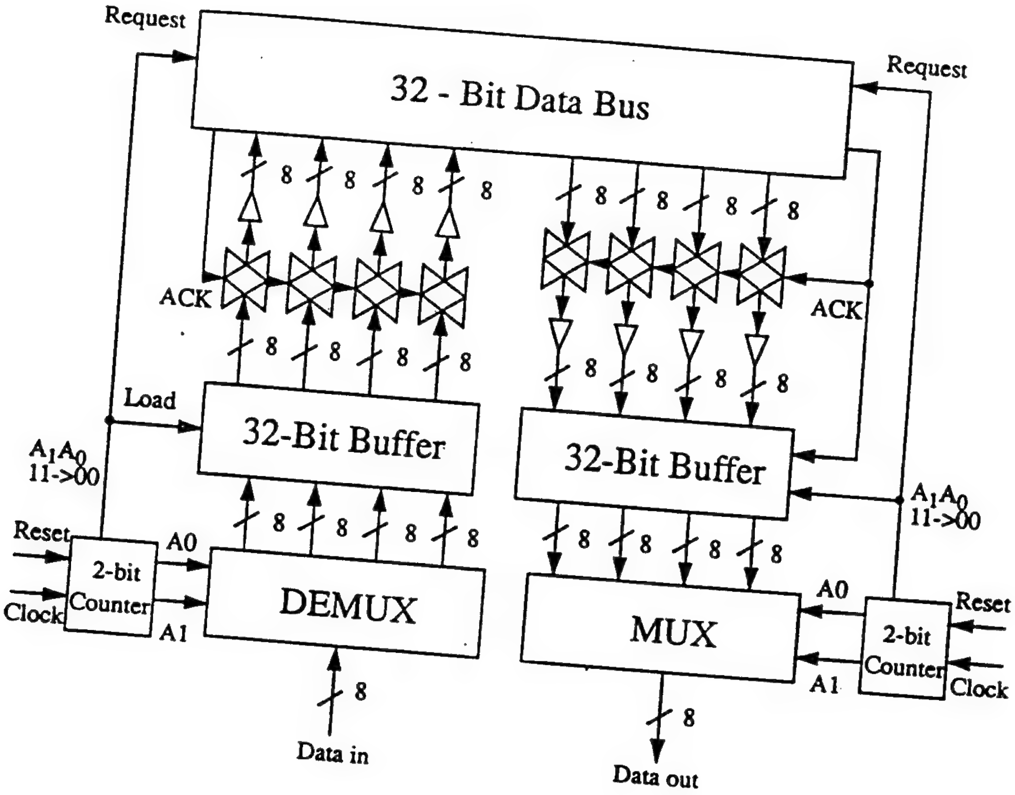


Figure 5-2
Block diagram of the I/O interface.

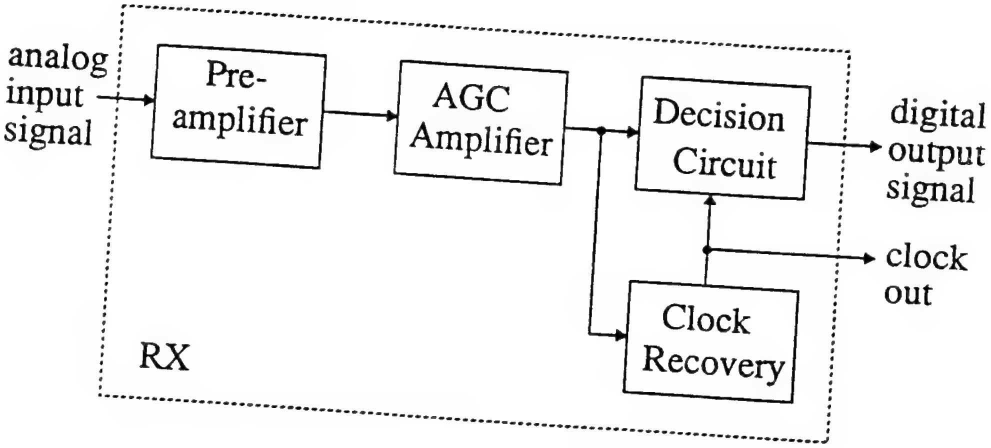


Figure 5-3
Block diagram of the receiver module.

A block diagram of the transmitter circuit is shown in Figure 5-4. It contains a driver circuit and a current source. The driver circuit converts digital binary signals into laser-diode driving currents.

The laser diode converts the driving currents into optical power output, which is transmitted through the optical fibers and free space.

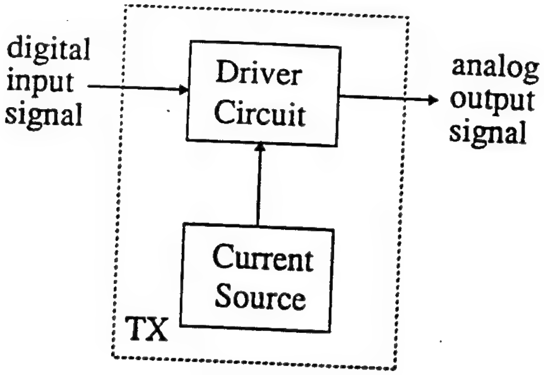


Figure 5-4
Block diagram of the transmitter module.

Vertical-cavity surface-emitting lasers (VCSELs) represent a novel class of semiconductor diode lasers with unique properties. Their main advantages over conventional edge-emitting lasers include wafer-scale processing without cleavage, inherent single-longitudinal-mode operation, and a nearly circular output beam with small divergence, which facilitates coupling to optical fibers or redirection and detection in free-space systems. The VCSEL geometry also offers a possibility of scaling the output power with the emitting area, which can potentially lead to high output power. Another very important feature of VCSELs, and one that is directly relevant to the proposed project, is their *compatibility with vertical stacking architectures*, which permits them to be integrated into more complex systems.

The single-element development of VCSELs can be regarded as a precondition for integration into densely-packed two-dimensional (2-D) arrays with an unconstrained arrangement of emitters. For example, large numbers of VCSELs can be monolithically integrated. These features make them ideal for many new applications, such as chip-to-chip communications, free-space optical communications, optical recording, etc. They are especially attractive for applications requiring a *high degree of parallelism*, such as optical intercommunications.

VCSELs are inherently low power consumption devices. The driving current is typically 1 mA, while the threshold voltage is about 1.5 V. Therefore, power dissipation is less than 2 mW. This power consumption level is much lower than other semiconductor diode lasers. Figure 5-5 shows a measured plot for voltage, current, output power, and power conversion efficiency. Clearly, the power dissipation is low, while the output optical power is high. The power conversion efficiency from electrical to optical is very high at low temperatures, and reasonably high (not shown) at room temperature. A maximum efficiency of over 25% can be achieved at room temperature. Therefore, if each laser consumes 2 mW of electrical power and emits about 500 μW of optical power, the maximum heat generated by each laser is about 1.5 mW. For interlayer optical interconnects, the required optical power is much lower than 500 μW, due to minimal losses in the lens and optical coupling. Thus, a power level of about 100 μW is enough for interlayer interconnect applications, and the heat generated by each VCSEL can be smaller than 1 mW. Lower threshold VCSELs are currently in development. Future technology should permit placement of more VCSELs in the same area, implying high-density laser array packaging.

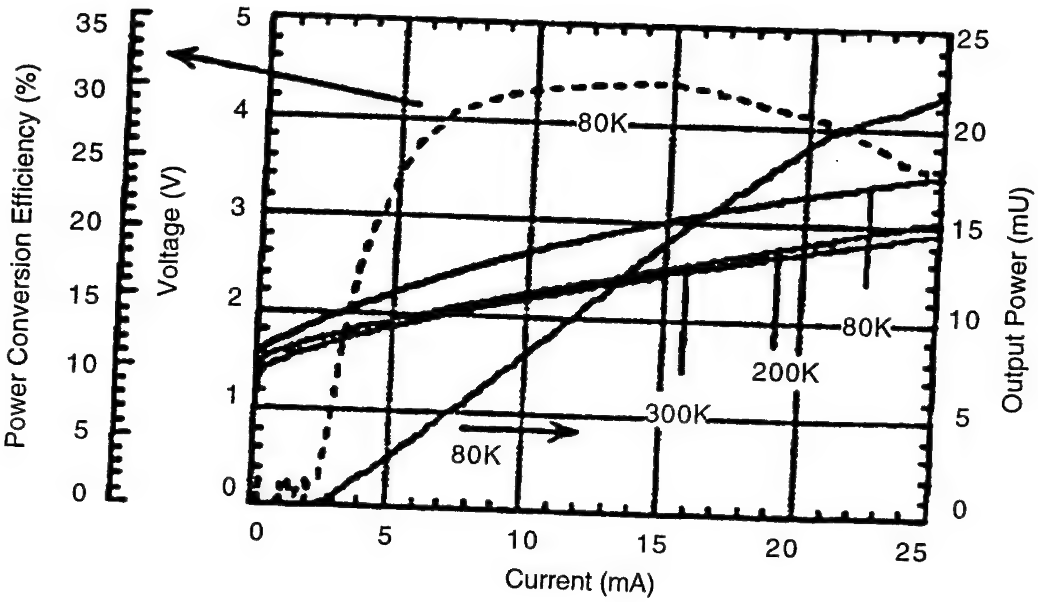


Figure 5-5
Measured VCSEL response. Low threshold voltage and driving current and high optical power output is clearly possible with the VCSEL. Electrical-to-optical power conversion efficiency is quite high.

An optical data link technology, based on vertical-cavity surface-emitting lasers (VCSEL) operating at low temperature, was developed, as shown in the photo in Figure 5-6. This VCSEL technique can operate at low temperatures and offers superior laser performance, including higher modulation speed, low power dissipation, and high power conversion efficiency, which can significantly exceed room temperature performance characteristics. For edge-emitting lasers, threshold current decreases, while slope efficiency increases monotonically with temperature. Modulation frequency response at a constant output power typically increases with decreasing temperature, when not limited by parasitic elements. These properties make it advantageous to operate lasers at a low temperature. VCSELs, on the other hand, typically exhibit parabolic dependence of threshold current on temperature, which results from thermally-induced de-tuning of the wavelengths of the lasing mode and gain peak, which shift with temperature at different rates. The consequences of misalignment are increased in operating current and voltage, which increase power dissipation at low temperatures. This can be circumvented by designing a VCSEL whose lasing mode and gain peak are intentionally de-tuned at room temperature, but which become aligned at a designated lower temperature.

PHOTO (Fig.2-4)
Final 1195.3327
F 29601-95-C-0097

(a)

PHOTO
Figure 2-9
F 1195.3327
F 29601-95-C-0097

(b)

Figure 5-6
(a) Picture of the purchased commercially available VCSEL array; (b) high speed (over 3 GHz) driver circuit for the VCSEL array element.

In addition to significantly improved lasing performance at low temperatures, VCSELs also offer good optical properties, such as small beam divergence and circular beam characteristics, which facilitate optical coupling through a fiber or free space. Properly designed VCSELs can maintain thermally stable lasing characteristics over a very wide range of temperatures, providing an optical interconnect that is stable against temperature variations. Figure 5-7 shows a setup, where a VCSEL was used in a free space interconnect. The VCSEL was mounted on a high-speed package. Electrical access to the VCSEL was achieved through a high-speed SMA launcher and a 50-ohm microstripline. The light produced by the mounted VCSEL propagates through free-space, and is coupled out of the optical window situated ~ 2 cm above the VCSEL's emitting surface. An external lens focuses this low divergence beam into the small optical aperture of a high-speed p-i-n photodetector, which is followed by a transimpedance amplifier. The VCSEL is digitally modulated by NRZ pseudo-random data from a pattern generator, with a word length of 2^{23}-1 bits and a data rate of up to 2 Gb/s. The optical output of the VCSEL is detected by the p-i-n detector and the transimpedance amplifier, and the eye diagrams are recorded by a communication signal analyzer.

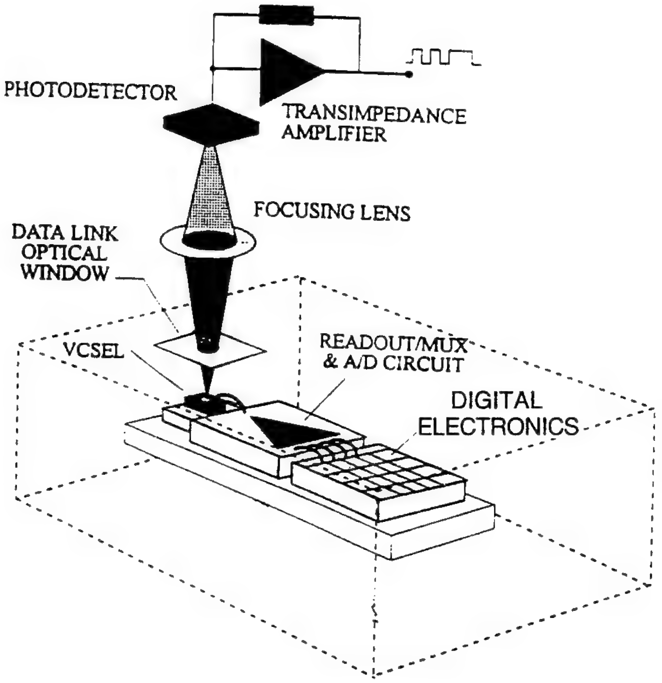


Figure 5-7
Configuration for the characterization of a high-speed optical link, using an optimized VCSEL as the optical source and a free-space optical interconnect approach to direct light into an external photodetector and a transimpedance amplifier.

5.3 Photonic Interconnect Chip Design with Digital Processor

To develop a photonic interconnect for use with a digital image processor application, optical transceivers are required that can convert a TTL electronic signal to an analog optical signal and convert the analog optical signal back to a TTL electronic signal. Since our applications are in parallel-bit communications, an array of optical transceivers are required in a node. Figure 5-8 shows a schematic of an optical transceiver board design. A high-density electrical connector is mounted on the board, providing the interface between the transceiver and a standard electronic module with digital I/O ports. The functions of the connector pins may include TTL parallel data/address in and parallel data/address out, clock, sync enable, and some control and status signals.

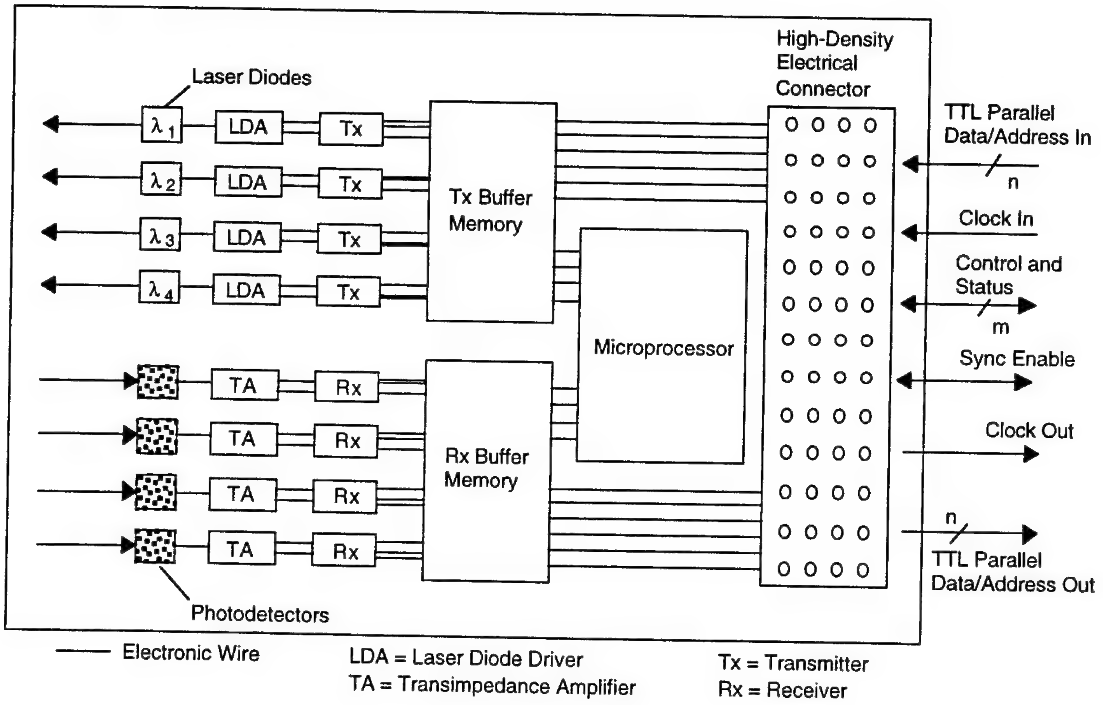


Figure 5-8
Optical transceiver design in a communication node.

Laser diodes can be mounted in a receptacle with a GRIN lens for beam collimation. The PIN photodetector is also mounted in a receptacle and uses a standard optical lens for beam focusing.

The transmitter chip accepts a parallel data word (e.g., 8-bit or 16-bit data), encodes it into symbol coding (using 4B/5B), converts it to a serial stream (via a parallel-to-serial converter), performs an NRZ to NRZI conversion, and outputs the data across a serial ECL interface. Similarly, the receiver chip performs the inverse function of the transmitter. It accepts a serial diffracted ECL signal, recovers clock and data, performs an NRZI to NRZ conversion, translates the resulting 5B symbols to 4B data, assembles the parallel data (via a serial-to-parallel converter), and outputs it to the electronic system. The laser diode driver chip accepts the incoming differential ECL signal from the transmitter, and converts it at high speed into a current driver for the laser diode. At the same time, the laser diode driver chip also provides flexible driving current bias and modulation control input to simplify overall transceiver design. Similarly, the transimpedance amplifier chip accepts a signal from the PIN photodetector, converts it into a high-speed differential ECL signal to the receiver and provides several additional lines for biasing or gain control.

The purpose of the photonic interface chip is to provide efficient electronic-to-optical and optical-to-electronic conversion. The chip can be interfaced directly with processor, memory, and I/O chips, via a local bus. Figure 5-9(a) illustrates this basic operation. A processing node (consisting of processor, memory, and I/O chips) in layer i communicates with another processor node in the adjacent layer i + 1 via an optical interconnect bus. The combination of the optical interconnection bus and the photonic interface chip provides high-throughput and low-latency data communication. Thus, from a logical point of view, the two processor nodes (in different layers) communicate with each other through the extension of their local bus (see Figure 5-9(b)). In other words, the optical interconnect bus and the photonic interface chip provide a coherent communication data link between nodes in different layers regardless of separation distance. The nodes act as if they were adjacent to each other on the same board.

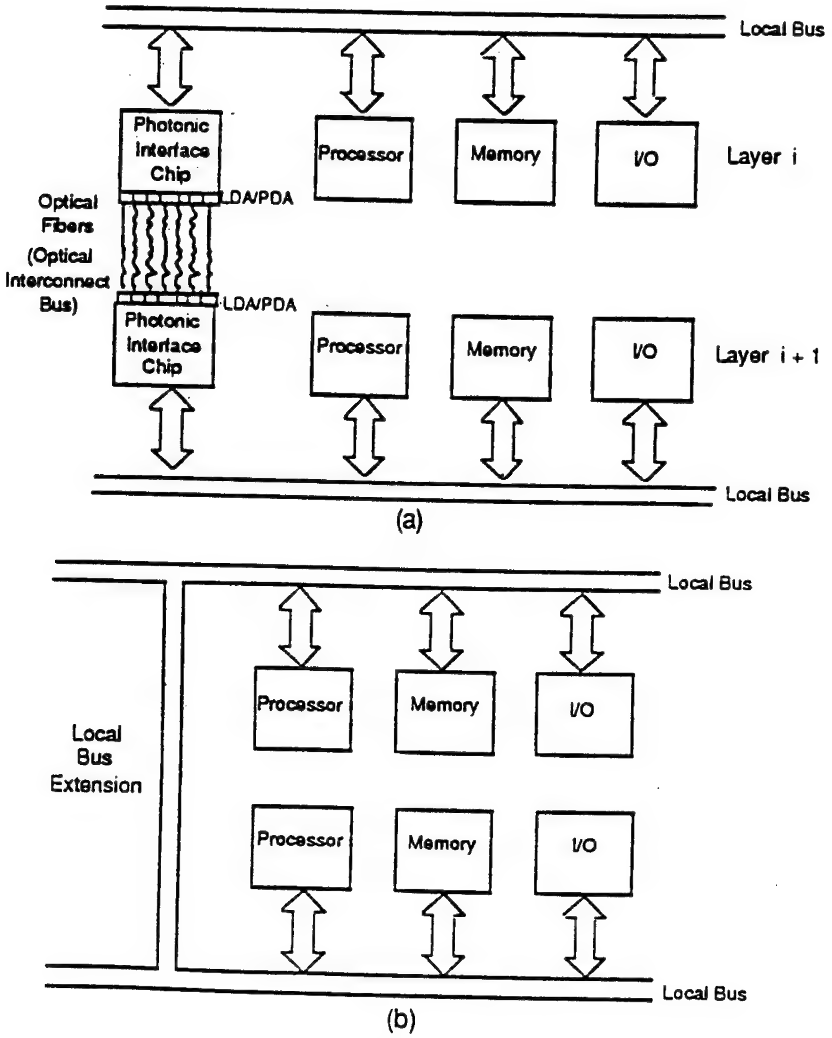


Figure 5-9
Photonic interface chip operation: (a) physical layout and (b) logic layout.
(Note: LDA = Laser Diode Array; PDA = Photodetector Array.)

5.4 System Application Study

In this section, we present a special design case that utilizes photonic interconnect for Focal Plane Array (FPA) image processing application. The 3-D packaging of the FPA sensor/processor system is schematically shown in Figure 5-10. The FPA photodetector array with small pixel spacing is fabricated and packaged with analog-to-digital (A/D) converter chips. Packaging is done with an Indium Solder bump technique. The FPA photodetectors are grouped into many clusters, each with 3×3 or up to 10×10 photodetectors, depending on image-processing speed requirements. Each A/D converter handles signal conversion for a cluster of FPA photodetectors. This greatly reduces the number of A/D converters, and reduces the difficulty of packaging FPA photodetectors with A/D converter chips. The resulting A/D converter outputs are sent through bonded wires to the electronic drivers of the vertical-cavity surface-emitting laser (VCSEL) arrays. FPA sensor units with VCSELs and laser drivers can be packaged by multichip module (MCM) technologies. The resulting board is called the FPA sensor board.

The downward vertically-emitted laser beams carrying the A/D-converted FPA sensor signals are coupled into the optical pipe array for beam collimation. The optical pipe array is fabricated on a separate board called an optical pipe array board, under the FPA sensor board. Both the FPA sensor board and the optical pipe array board are inside the dewar, which provides a cool environment for better FPA performance.

The planar optical interconnect board consists of many holographic optical elements. These holographic elements route the received collimated optical beam by diffraction to one or many planar locations on the board. Planar optical beam propagation is by total internal reflection inside a transparent board medium, such as optical glass. Beams are redirected to designated photodetectors at the processor board (again by holographic diffraction) and through another optical pipe array board. This optical pipe array board focuses the light beams to the photodetectors. Optical signal beams received at photodetectors on the processor board are converted to electronic format, and are then processed by the electronic multi-processing element array.

The processor board is packaged using advanced electronic packaging technologies. The planar optical interconnect board, the second optical pipe array board, and the processor board are all outside the dewar, since the processor board can be relatively hot, and the planar optical interconnect board and the second optical pipe array board must be close to the processor board for effective optical beam focusing and alignment. A side view of Figure 5-10 is shown in Figure 5-11.

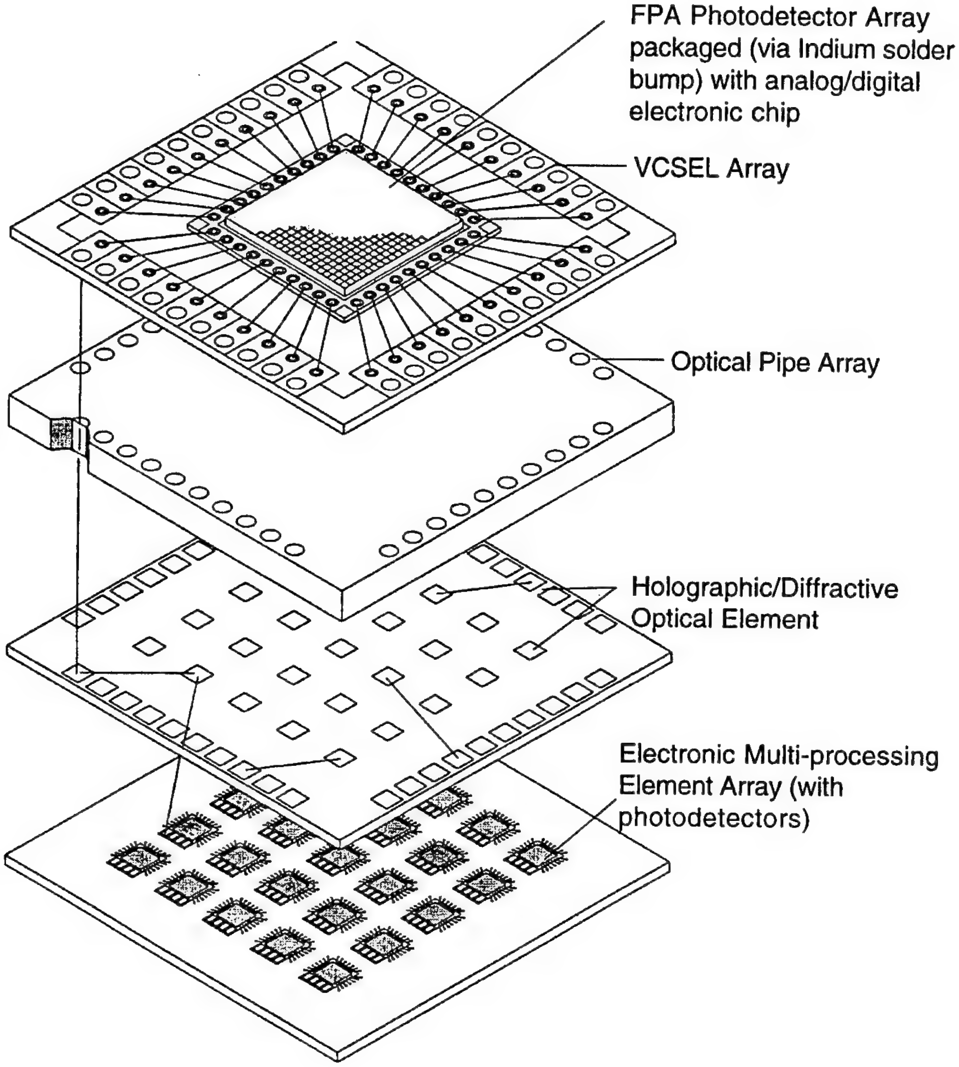

FPA Photodetector Array packaged (via Indium solder bump) with analog/digital electronic chip
VCSEL Array
Optical Pipe Array
Holographic/Diffractive Optical Element
Electronic Multi-processing Element Array (with photodetectors)

Figure 5-10
3-D packaging architecture for photonic interconnect-based FPA sensors and processors.

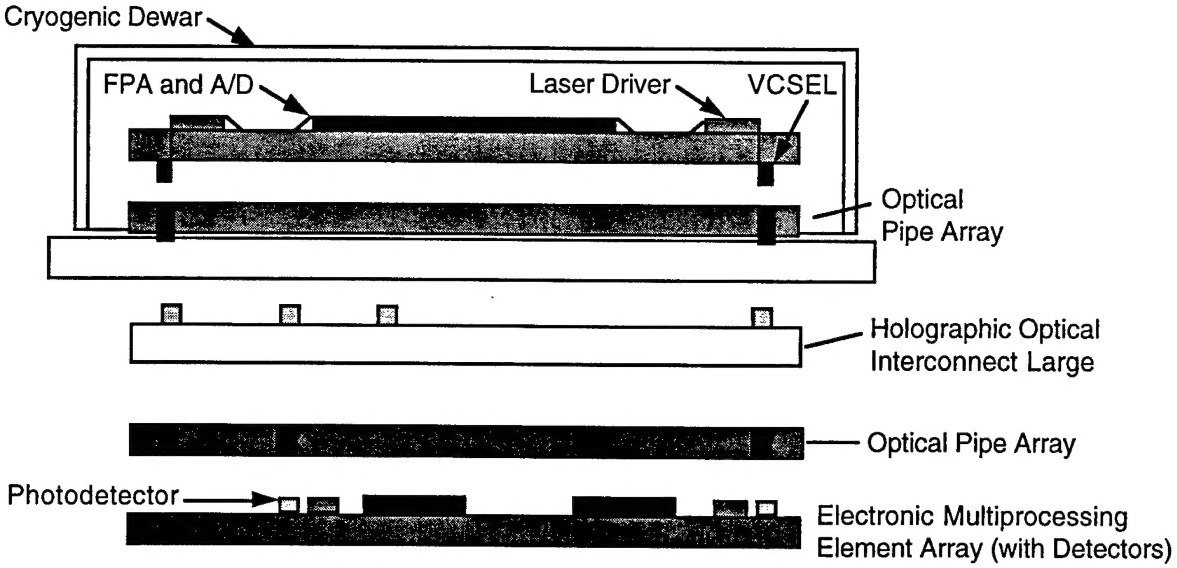

Cryogenic Dewar
FPA and A/D
Laser Driver
VCSEL
Optical Pipe Array
Holographic Optical Interconnect Large
Optical Pipe Array
Photodetector
Electronic Multiprocessing Element Array (with Detectors)

Figure 5-11
Side view of 3-D integration architecture for FPA sensor/ processor system.

When multiple processor boards are needed, 3-D processor packaging is used, as shown in Figure 5-12. On each board, there is a designated area for vertical optical communications (inset drawing), which includes optical pipe arrays, VCSEL arrays, and photodetector arrays. These inter-board interconnect areas are connected to processors by microstrip lines on MCM-packaged processor boards. Optical interconnect architecture allows communications between neighboring and/or non-neighboring boards.

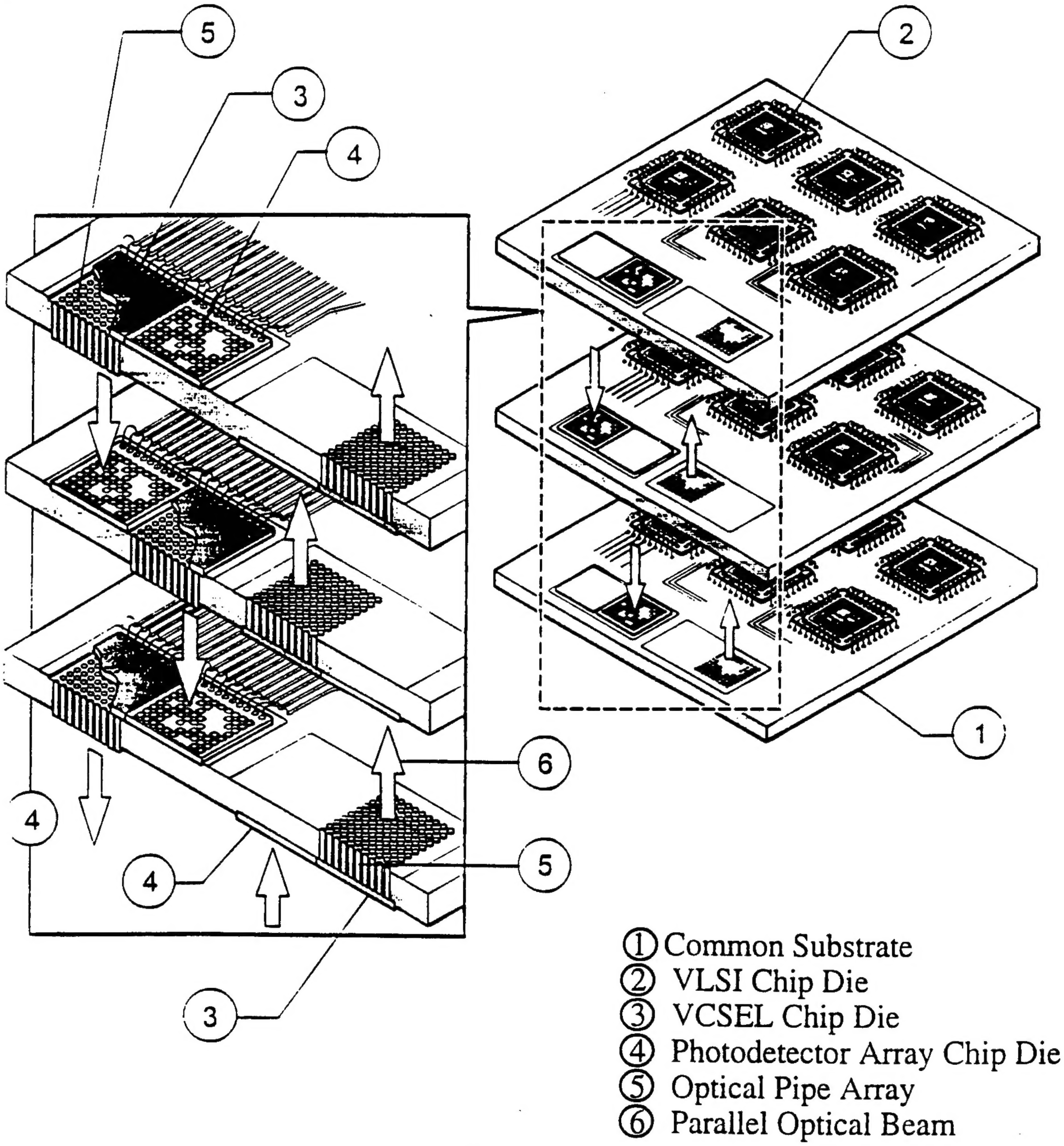


Figure 5-12
Architecture design for multi-processor board system by parallel photonics interconnects.

6.0 CONCLUSION

A CNN (Cellular Neural Network)-based 3-D array machine was explored in this project. This CNN machine was implemented with discrete circuit components and can perform image processing in real time. It is programmable and flexible. Different image processing functions can be achieved by changing CNN templates. In addition, since the CNN machine is operated in the analog domain, very short latency is required. This machine is suitable for front-end applications. It can be mounted onto other image processing equipment and increase performance instantly.

Based on our preliminary studies, we anticipate that this CNN-based front-end image processor will be very attractive for use with products which require high-speed video image processing. These related products include multimedia systems, animation, video games, and medical imaging. The projected markets and capturable market for our CNN product are shown in Table 6-1.

Table 6-1 Projected Commercial Markets and Market Capture for POC's Analog Image Processing Device

Year	Multimedia Products (30% Annual Market Growth)		Entertainment (20% Annual Market Growth)		Medical Imaging (20% Annual Market Growth)	
	Market	% Capture	Market	% Capture	Market	% Capture
1998	165	0	100	0	68	0
1999	215	0	120	0	82	2 %
2000	280	2 %	144	2 %	98	5 %
2001	363	5 %	173	5 %	118	8 %
2002	470	10 %	207	10 %	141	10 %

Because of time-to-market constraints and market acceptance, this prediction assumes a very conservative market capture in the first five years. However, because these markets are still enormous, we expect strong growth is possible for this product.

In conclusion, this developed CNN machine has great potential commercial applicability. In addition, this system can also be used in military applications to increase the throughput and analysis for future digitized battlefield video images.

7.0 REFERENCES

1. J.A. Anderson, An Introduction to Neural Networks, MIT Press, 1995.
2. L.O. Chua and L. Yang, "Cellular neural network: theory and applications," IEEE Trans. On Circuit and System, Vol. 35, Oct., 1988.
3. J.A. Fried, "Optical I/O for high speed CMOS systems," Optical Engineering, Vol. 25, Oct., 1986.
4. P.R. Haugen, S. Rychnovsky, A. Husain, and L. Hutcheson, "Optical interconnects for high speed computing," Optical Engineering, Vol. 25, Oct., 1986.

5. M. Kilcoyne, S. Beccue, K. Pedrotti, R. Asatourian and R. Anderson, "Optoelectronic integrated circuits for high speed signal processing," Optical Engineering, Vol. 25, Oct., 1986.
6. R. Piedra and A. Frittsch, "Digital signal processing comes of age," IEEE Spectrum, May, 1996.
7. T. Roska and L.O. Chua, " The CNN universal machine: an analogic array computer," IEEE Trans. On Circuit and System, Vol. 40, March, 1993.
8. B. Sheu and J. Choi, Neural Information Processing and VLSI, Kluwer Academic Publishers, Boston, MA, 1995.